Exploring Concepts of HyperFuzzy, HyperNeutrosophic, and HyperPlithogenic Sets (II), pp. 91-151, in Takaaki Fujita, Florentin Smarandache: *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond. Fourth volume: HyperUncertain Set (Collected Papers).* Gallup, NM, United States of America – Guayaquil (Ecuador): NSIA Publishing House, 2025, 314 p.

Chapter 2

Exploring Concepts of HyperFuzzy, HyperNeutrosophic, and HyperPlithogenic Sets II

Takaaki Fujita[1]*, Florentin Smarandache[2]

[1]* Independent Researcher, Shinjuku, Shinjuku-ku, Tokyo, Japan. t171d603@gunma-u.ac.jp

[2] University of New Mexico, Gallup Campus, NM 87301, USA. smarand@unm.edu

Abstract. This paper delves into the advancements of classical set theory to address the complexities and uncertainties inherent in real-world phenomena. It highlights three major extensions of traditional set theory—Fuzzy Sets [288], Neutrosophic Sets [237], and Plithogenic Sets [243]—and examines their further generalizations into Hyperfuzzy [106], HyperNeutrosophic [90], and Hyperplithogenic Sets [90].

Building on previous research [83], this study explores the potential applications of HyperNeutrosophic Sets and SuperHyperNeutrosophic Sets across various domains. Specifically, it extends fundamental concepts such as Neutrosophic Logic, Cognitive Maps, Graph Neural Networks, Classifiers, and Triplet Groups through these advanced set structures and briefly analyzes their mathematical properties.

Keywords: Fuzzy set, Neutrosophic set, Hyperstructure, Hyperfuzzy set, Hyperneutrosophic set

1. Introduction

This paper is closely related to [83]. Readers are encouraged to review [83] in advance, as needed.

1.1. *Fuzzy Sets, Neutrosophic Sets, and Plithogenic Sets*

Set theory, a cornerstone of mathematics, provides a framework for analyzing collections of elements called "sets" [61,139]. This study examines three major extensions—Fuzzy Sets [288], Neutrosophic Sets [237], and Plithogenic Sets [243]—and their generalizations into Hyperfuzzy [106], HyperNeutrosophic [90], and Hyperplithogenic Sets [90].

These frameworks address various dimensions of uncertainty. *Fuzzy Sets* represent imprecision through membership values between 0 and 1 [288]. *Neutrosophic Sets* enhance this by adding truth, indeterminacy, and falsity components, offering richer analyses of complex systems [237]. *Plithogenic Sets* further extend these ideas to handle multidimensional uncertainty and contradictions, making them particularly effective for analyzing highly complex systems [244, 256].

1.2. *Hyperfuzzy, HyperNeutrosophic, and Hyperplithogenic Sets*

Extensions of Fuzzy Sets [90, 106, 144, 262], Neutrosophic Sets [90], Plithogenic Sets [90], Soft Sets [1, 84, 99, 121, 137, 213, 226, 229, 242, 250], Rough Sets [90], and Vague Sets [90] have been developed using Hyperstructures and n-SuperHyperstructures.

For instance, Fuzzy Sets have been extended into Hyperfuzzy Sets [80, 106, 144–147, 172, 176, 177, 190, 262] and SuperHyperfuzzy Sets [90]. Similarly, Neutrosophic Sets have been extended into HyperNeutrosophic Sets [90, 241] and SuperHyperNeutrosophic Sets [90, 241], while Plithogenic Sets have been extended into HyperPlithogenic Sets [90, 241] and SuperHyperPlithogenic Sets [90, 241].

1.3. *Our Contribution in This Paper*

This section highlights the contributions of this paper. Building on previous research [83], we investigate the potential applications of HyperNeutrosophic Sets and SuperHyperNeutrosophic Sets in various domains.

The study focuses primarily on theoretical exploration and mathematical formulation. For example, we extend concepts such as Neutrosophic Logic, Cognitive Maps, Graph Neural Networks, Classifiers, and Triplet Groups using HyperNeutrosophic Sets and SuperHyperNeutrosophic Sets, and briefly analyze their properties.

Future research should include experimental validation and application-oriented studies to facilitate practical implementation in specific fields. Through this work, we aim to advance this area of study and encourage further exploration and development of related topics.

2. **Preliminaries and Definitions**

This section outlines the essential concepts and definitions necessary for understanding the discussions in this paper. While we aim to present the fundamental ideas concisely, a comprehensive exploration of all related terms lies beyond the scope of this work. Readers are encouraged to consult the cited references for a more in-depth understanding.

2.1. *Basics of Set Theory and Others*

This subsection provides a brief overview of foundational principles in set theory. For a detailed discussion, we recommend standard references such as [123, 139, 143].

Definition 2.1 (Set)**.** [139] A *set* is a well-defined collection of distinct objects, referred to as *elements*. For any object x, it is always determinable whether x is an element of a given set. If x belongs to a set A, this is denoted as $x \in A$. Sets are often represented using curly braces. For example, $A = \{1, 2, 3\}$ represents a set containing the elements 1, 2, and 3.

Definition 2.2 (Subset)**.** [139] A set A is called a *subset* of another set B, written as $A \subseteq B$, if every element of A is also an element of B. This relationship is formally expressed as:

$$A \subseteq B \iff \forall x \, (x \in A \implies x \in B).$$

If $A \subseteq B$ and $A \neq B$, A is referred to as a *proper subset* of B, denoted by $A \subset B$.

Definition 2.3 (Empty Set)**.** [139] The *empty set*, denoted as $\emptyset$, is the unique set containing no elements. It is formally defined as:

$$\forall x \, (x \notin \emptyset).$$

For example, the empty set can be represented as $\emptyset = \{\}$.

Definition 2.4 (Universal Set)**.** [139] The *universal set*, denoted by U, represents the set containing all objects under consideration within a specific context. Any set A under analysis is a subset of U. Formally:

$$A \subseteq U \quad \text{for any set } A.$$

Although some concepts may not have a direct connection to set theory, the following fundamental mathematical definitions will also be employed. As these are basic definitions, readers are encouraged to refer to relevant literature as needed.

Definition 2.5 (Real Numbers)**.** (cf. [70,127]) The set of real numbers, denoted by $\mathbb{R}$, includes all rational and irrational numbers, which can be represented as points on the real number line. Examples are integers, fractions, and roots.

Definition 2.6 (Natural Numbers)**.** (cf. [160]) The set of natural numbers, denoted by $\mathbb{N}$, consists of all positive integers starting from 1:

$$\mathbb{N} = \{1, 2, 3, \ldots\}.$$

Some conventions also include 0, depending on the context.

Definition 2.7 (Homomorphism)**.** (cf. [75, 220]) Let $(A, \star)$ and $(B, \circ)$ be two algebraic structures. A *homomorphism* is a function $f : A \to B$ that satisfies:

$$f(a \star a') = f(a) \circ f(a') \quad \text{for all } a, a' \in A.$$

Definition 2.8 (Operation)**.** [139] An *operation* is a function or rule that combines elements of a set S to produce another element within S. Formally, an operation $\circ$ on S is defined as:

$$\circ : S \times S \to S.$$

Examples include addition and multiplication, which are operations on the set of real numbers $\mathbb{R}$.

Definition 2.9 (Binary Operation)**.** [34] A *binary operation* on a set S is a function $* : S \times S \to S$ that combines two elements $a, b \in S$ to produce another element $a * b \in S$. Examples include addition and subtraction, both of which are binary operations on $\mathbb{R}$.

Definition 2.10 (Graph)**.** [64–66] A *graph*, denoted $G = (V, E)$, consists of:

- V: A set of vertices (or nodes).
- E: A set of edges, where each edge is an unordered pair of vertices $\{u, v\}$, $u, v \in V$.

Definition 2.11 (Directed Graph)**.** (cf. [22,66]) A *directed graph*, denoted $G = (V, E)$, consists of:

- V: A set of vertices (or nodes).
- E: A set of directed edges, where each edge is an ordered pair of vertices (u, v), $u, v \in V$.

Definition 2.12 (Matrix)**.** [25] A matrix is a rectangular array of elements arranged in rows and columns, typically denoted as $A = [a_{ij}]$, where a_{ij} represents the element in the i-th row and j-th column.

Definition 2.13 (Adjacency Matrix of a Graph)**.** (cf. [114]) Let $G = (V, E)$ be a graph with vertex set V and edge set E. The adjacency matrix of G, denoted as $A = [a_{ij}]$, is a square matrix of size $|V| \times |V|$, where $|V|$ is the number of vertices in G. Each entry a_{ij} is defined as:

$$a_{ij} = \begin{cases} 1, & \text{if there is an edge from vertex } v_i \text{ to } v_j, \\ 0, & \text{otherwise.} \end{cases}$$

For undirected graphs, the adjacency matrix A is symmetric, whereas for directed graphs, A may not be symmetric.

Definition 2.14 (Weight Matrix)**.** (cf. [193, 268]) A weight matrix is a matrix where each element represents a weight or parameter, often used to describe relationships in graphs, neural networks, or optimization problems, such as edge weights or neural connection strengths.

Definition 2.15 (Approximation)**.** (cf. [206]) Approximation refers to the process of representing a function or a value by another function or value that is close to the original within a specified level of accuracy. It is fundamental in numerical analysis and machine learning.

Definition 2.16 (Vector)**.** (cf. [63]) A vector is an ordered tuple of elements, typically from a field $\mathbb{R}$ or $\mathbb{C}$, representing a point in an n-dimensional space or a directed quantity with both magnitude and direction.

2.2. *Hyperstructure and Superhyperstructure*

This subsection introduces the concepts of Hyperstructure and Superhyperstructure, advanced mathematical frameworks designed to represent hierarchical and multi-layered systems. A *Hyperstructure* builds upon the powerset of a base set to model relationships within collections of elements. Extending this notion, a *Superhyperstructure* utilizes the n-th powerset to represent intricate hierarchical systems across multiple layers [81, 82, 254, 255]. Below, we formalize the n-th powerset and its related constructs.

Definition 2.17 (Base Set)**.** A *base set* is the foundational set S from which powersets and hyperstructures are constructed. Formally:

$$S = \{x \mid x \text{ is an element within the specified domain}\}.$$

All subsets and operations within $\mathcal{P}(S)$ or $\mathcal{P}_n(S)$ are derived from the elements of S.

Definition 2.18 (Powerset)**.** [86, 215] The *powerset* of a set S, denoted as $\mathcal{P}(S)$, is the collection of all subsets of S, including the empty set and S itself:

$$\mathcal{P}(S) = \{A \mid A \subseteq S\}.$$

Definition 2.19 (n-th Powerset)**.** (cf. [86, 235, 254])

The n-th powerset of a set H, denoted $P_n(H)$, is defined recursively. Starting with the standard powerset, the construction proceeds as:

$$P_1(H) = P(H), \quad P_{n+1}(H) = P(P_n(H)), \quad \text{for } n \geq 1.$$

The n-th non-empty powerset, denoted $P_n^*(H)$, excludes the empty set:

$$P_1^*(H) = P^*(H), \quad P_{n+1}^*(H) = P^*(P_n^*(H)).$$

Here, $P^*(H)$ is the powerset of H excluding the empty set.

To formalize the concepts of Hyperstructure and Superhyperstructure, we proceed with the following definitions.

Definition 2.20 (Classical Structure)**.** (cf. [235,254]) A *Classical Structure* is a mathematical framework defined on a non-empty set H equipped with one or more *Classical Operations* that satisfy specific axioms. A *Classical Operation* is a function:

$$\#_0 : H^m \to H,$$

where $m \geq 1$ and H^m represents the m-fold Cartesian product of H. Examples include addition and multiplication in algebraic structures like groups and rings.

Definition 2.21 (Hyperstructure)**.** (cf. [86, 235, 254]) A *Hyperstructure* extends a Classical Structure by operating on the powerset of a base set. Formally:

$$\mathcal{H} = (\mathcal{P}(S), \circ),$$

where S is the base set, $\mathcal{P}(S)$ is its powerset, and $\circ$ is an operation defined on subsets of $\mathcal{P}(S)$.

Definition 2.22 (n-Superhyperstructure)**.** (cf. [235, 254]) An *n-Superhyperstructure* generalizes a Hyperstructure by utilizing the n-th powerset of a base set. It is defined as:

$$\mathcal{SH}_n = (\mathcal{P}_n(S), \circ),$$

where S is the base set, $\mathcal{P}_n(S)$ is the n-th powerset of S, and $\circ$ is an operation on elements of $\mathcal{P}_n(S)$.

A representative example of a superhyperstructure is the *SuperHypergraph*, which incorporates advanced elements such as superedges and supervertices, offering a more abstract and versatile framework for hierarchical modeling [38, 85, 90, 91, 93–96, 105, 116, 117, 183, 246, 247, 249, 252, 254]. Additionally, concepts such as *SuperHyperfunction* have also been explored in the literature [251, 253].

2.3. *Fuzzy Set, Hyperfuzzy Set, and Superhyperfuzzy Set*

This subsection presents the formal definitions of Fuzzy Set, Hyperfuzzy Set, and Superhyperfuzzy Set. These concepts extend the traditional notion of fuzzy values into hierarchical structures, offering more refined tools for representing uncertainty.

Definition 2.23 (Fuzzy Set)**.** [288–292] A *fuzzy set* τ in a non-empty universe Y is a mapping $\tau : Y \to [0, 1]$. A *fuzzy relation* on Y is a fuzzy subset δ of $Y \times Y$. If τ is a fuzzy set in Y and δ is a fuzzy relation on Y, δ is called a *fuzzy relation on* τ if:

$$\delta(y, z) \leq \min\{\tau(y), \tau(z)\} \quad \text{for all } y, z \in Y.$$

Example 2.24 (Fuzzy Set: Membership in a Fitness Club)**.** Consider the universe $Y = \{\text{John, Alice, Bob, Sarah}\}$, representing a group of people. A fuzzy set τ defines their membership in a fitness club based on their participation level, where:

$$\tau(y) = \begin{cases} 1.0, & \text{if the person is a regular member (e.g., Alice),} \\ 0.8, & \text{if the person participates occasionally (e.g., Bob),} \\ 0.3, & \text{if the person rarely participates (e.g., Sarah),} \\ 0.0, & \text{if the person is not a member (e.g., John).} \end{cases}$$

The mapping $\tau : Y \to [0, 1]$ intuitively represents the degree of belonging for each individual in the club.

Definition 2.25 (Hyperfuzzy Set)**.** [29, 106, 144, 189, 262] Let X be a non-empty set. A *hyperfuzzy set* over X is defined as a mapping $\tilde{\mu} : X \to \tilde{P}([0,1])$, where $\tilde{P}([0,1])$ represents the set of all non-empty subsets of the interval $[0,1]$.

Example 2.26 (Hyperfuzzy Set: Customer Satisfaction Ratings)**.** Customer satisfaction ratings are often analyzed from the perspective of fuzzy set theory [76, 76, 135, 166]. Consider a set $X = \{\text{Product A, Product B, Product C}\}$, representing three products. A hyperfuzzy set $\tilde{\mu}$ maps each product to a set of customer satisfaction ratings, where:

$$\tilde{\mu}(x) = \begin{cases} \{0.9, 1.0\}, & \text{for Product A (highly satisfied customers),} \\ \{0.4, 0.6, 0.8\}, & \text{for Product B (moderately satisfied customers),} \\ \{0.1, 0.2\}, & \text{for Product C (low satisfaction levels).} \end{cases}$$

Here, $\tilde{\mu} : X \to \tilde{P}([0,1])$, where each product is associated with a set of satisfaction levels, capturing the diverse opinions of customers.

Definition 2.27 (n-SuperHyperFuzzy Set)**.** [90, 241] Let X be a non-empty set. An n-SuperHyperFuzzy Set is a recursive extension of fuzzy sets, hyperfuzzy sets, and superhyperfuzzy sets, defined as:

$$\tilde{\mu}_n : \tilde{\mathcal{P}}_n(X) \to \tilde{\mathcal{P}}_n([0,1]),$$

where:

- $\tilde{\mathcal{P}}_1(X) = \tilde{\mathcal{P}}(X)$, and for $k \geq 2$,
$$\tilde{\mathcal{P}}_k(X) = \tilde{\mathcal{P}}(\tilde{\mathcal{P}}_{k-1}(X)),$$
represents the k-th nested family of non-empty subsets of X.
- $\tilde{\mathcal{P}}_n([0,1])$ is similarly defined for the interval $[0,1]$.
- $\tilde{\mu}_n$ maps each element $A \in \tilde{\mathcal{P}}_n(X)$ to a non-empty subset $\tilde{\mu}_n(A) \subseteq [0,1]$, which represents the membership degrees of A at the n-th hierarchical level.

2.4. *Neutrosophic, HyperNeutrosophic, and SuperHyperNeutrosophic Sets*

Neutrosophic Sets enhance Fuzzy Sets by incorporating the concept of indeterminacy, allowing them to model situations that are neither entirely true nor false [237]. This framework offers a more comprehensive approach to handling real-world scenarios characterized by significant uncertainty and complexity, making it a focus of extensive research [100, 101, 155, 236, 238, 240, 248, 257, 258, 260, 261]. The formal definitions are provided below.

Definition 2.28 (Neutrosophic Set)**.** [237] Let X be a non-empty set. A *Neutrosophic Set* A on X is defined by three membership functions:

$$T_A : X \to [0,1], \quad I_A : X \to [0,1], \quad F_A : X \to [0,1],$$

where $T_A(x)$, $I_A(x)$, and $F_A(x)$ represent the degrees of truth, indeterminacy, and falsity for each $x \in X$. These values satisfy the condition:

$$0 \leq T_A(x) + I_A(x) + F_A(x) \leq 3.$$

Example 2.29 (Neutrosophic Set: Decision-Making in Hiring)**.** Decision-making is often studied in conjunction with Neutrosophic Sets [6, 11, 52, 60, 164, 181, 198, 286]. Consider $X = \{\text{Candidate A, Candidate B, Candidate C}\}$, representing applicants for a job. A Neutrosophic Set A defines the suitability of each candidate, where:

$$T_A(x), I_A(x), F_A(x)$$

denote the degrees of truth (suitability), indeterminacy (uncertainty), and falsity (unsuitability) for each candidate:

$$T_A(\text{Candidate A}) = 0.8, \quad I_A(\text{Candidate A}) = 0.1, \quad F_A(\text{Candidate A}) = 0.1,$$

$$T_A(\text{Candidate B}) = 0.5, \quad I_A(\text{Candidate B}) = 0.4, \quad F_A(\text{Candidate B}) = 0.1,$$

$$T_A(\text{Candidate C}) = 0.3, \quad I_A(\text{Candidate C}) = 0.2, \quad F_A(\text{Candidate C}) = 0.5.$$

Here, the Neutrosophic Set models the hiring committee's confidence, uncertainty, and rejection levels for each applicant.

Definition 2.30 (HyperNeutrosophic Set)**.** [90] Let X be a non-empty set. A *HyperNeutrosophic Set* on X is a mapping $\tilde{\mu} : X \to \tilde{P}([0,1]^3)$, where $\tilde{P}([0,1]^3)$ is the family of all non-empty subsets of the unit cube $[0,1]^3$. For each $x \in X$, $\tilde{\mu}(x) \subseteq [0,1]^3$ represents a collection of membership values, with each element comprising degrees of truth (T), indeterminacy (I), and falsity (F). These components satisfy:

$$0 \leq T + I + F \leq 3.$$

Example 2.31 (HyperNeutrosophic Set: Product Feedback Analysis)**.** Consider $X = \{\text{Product X, Product Y, Product Z}\}$, representing three products. A HyperNeutrosophic Set $\tilde{\mu}$ maps each product to a set of customer opinions, where each opinion is a triple (T, I, F) representing truth (positive feedback), indeterminacy (uncertainty), and falsity (negative feedback):

$$\tilde{\mu}(\text{Product X}) = \{(0.9, 0.1, 0.0), (0.8, 0.2, 0.0)\},$$

$$\tilde{\mu}(\text{Product Y}) = \{(0.6, 0.3, 0.1), (0.5, 0.4, 0.1), (0.7, 0.2, 0.1)\},$$

$$\tilde{\mu}(\text{Product Z}) = \{(0.4, 0.5, 0.1), (0.3, 0.6, 0.1)\}.$$

This representation captures the diversity of customer feedback, with multiple sets of opinions reflecting varying perspectives.

Definition 2.32 (n-SuperHyperNeutrosophic Set)**.** [90] Let X be a non-empty set. An n-SuperHyperNeutrosophic Set is a recursive extension of Neutrosophic and HyperNeutrosophic Sets, defined as:

$$\tilde{A}_n : \tilde{\mathcal{P}}_n(X) \to \tilde{\mathcal{P}}_n([0,1]^3),$$

where:

- $\tilde{\mathcal{P}}_1(X) = \tilde{\mathcal{P}}(X)$, and for $k \geq 2$,

$$\tilde{\mathcal{P}}_k(X) = \tilde{\mathcal{P}}(\tilde{\mathcal{P}}_{k-1}(X)),$$

  denotes the k-th nested family of non-empty subsets of X.
- $\tilde{\mathcal{P}}_n([0,1]^3)$ is defined analogously for the unit cube $[0,1]^3$.
- The mapping $\tilde{A}_n$ assigns to each $A \in \tilde{\mathcal{P}}_n(X)$ a subset $\tilde{A}_n(A) \subseteq [0,1]^3$, representing the degrees of truth (T), indeterminacy (I), and falsity (F) for A at the n-th hierarchical level.

For each $A \in \tilde{\mathcal{P}}_n(X)$ and $(T, I, F) \in \tilde{A}_n(A)$, the following condition holds:

$$0 \leq T + I + F \leq 3.$$

2.5. *HyperPlithogenic Set*

The Plithogenic Set extends traditional set theories, such as Neutrosophic and Fuzzy Sets, by incorporating multi-dimensional attributes and contradictions [243,244]. Below, we present its formal definition.

Definition 2.33 (Plithogenic Set)**.** [243, 244] Let S be a universal set, and $P \subseteq S$. A *Plithogenic Set* PS is defined as:

$$PS = (P, v, Pv, pdf, pCF),$$

where:

- v: an attribute.
- Pv: the set of possible values for the attribute v.
- $pdf : P \times Pv \to [0,1]^s$: the *Degree of Appurtenance Function (DAF)*, mapping elements and attribute values to a membership degree.
- $pCF : Pv \times Pv \to [0,1]^t$: the *Degree of Contradiction Function (DCF)*, quantifying contradictions between attribute values.

These functions satisfy the following axioms:

(1) *Reflexivity of DCF*:

$$pCF(a, a) = 0, \quad \text{for all } a \in Pv.$$

(2) *Symmetry of DCF*:

$$pCF(a,b) = pCF(b,a), \quad \text{for all } a,b \in Pv.$$

Example 2.34 (Examples of Plithogenic Sets)**.** [87, 98] The Plithogenic Set has various special cases:

- If $s = t = 1$, the set is called a *Plithogenic Fuzzy Set*.
- If $s = 2, t = 1$, it becomes a *Plithogenic Intuitionistic Fuzzy Set*.
- If $s = 3, t = 1$, it is referred to as a *Plithogenic Neutrosophic Set*.

Definition 2.35 (HyperPlithogenic Set)**.** [90] Let X be a non-empty set, and A a set of attributes. For each $v \in A$, let Pv be the range of possible values of v. A *HyperPlithogenic Set* HPS on X is defined as:

$$HPS = (P, \{v_i\}_{i=1}^n, \{Pv_i\}_{i=1}^n, \{\tilde{pdf}_i\}_{i=1}^n, pCF),$$

where:

- $P \subseteq X$: a subset of the universe.
- For each $v_i \in A$, Pv_i: the set of possible values for v_i.
- $\tilde{pdf}_i : P \times Pv_i \to \tilde{P}([0,1]^s)$: the *Hyper Degree of Appurtenance Function (HDAF)*, assigning membership degrees as sets.
- $pCF : \bigcup_{i=1}^n Pv_i \times \bigcup_{i=1}^n Pv_i \to [0,1]^t$: the *Degree of Contradiction Function (DCF)*.

Definition 2.36 (n-SuperHyperPlithogenic Set)**.** [90] Let X be a non-empty set, and let $V = \{v_1, v_2, \ldots, v_n\}$ be a set of attributes with respective ranges P_{v_i}. An n-SuperHyperPlithogenic Set $SHPS_n$ is defined recursively as:

$$SHPS_n = (P_n, V, \{P_{v_i}\}_{i=1}^n, \{\tilde{pdf}_i^{(n)}\}_{i=1}^n, pCF^{(n)}),$$

where:

- $P_1 \subseteq X$, and for $k \geq 2$,

  $$P_k = \tilde{\mathcal{P}}(P_{k-1}),$$

  representing the k-th nested family of subsets.
- For each v_i, P_{v_i}: the set of possible values of v_i.
- $\tilde{pdf}_i^{(n)} : P_n \times P_{v_i} \to \tilde{\mathcal{P}}([0,1]^s)$: the *HDAF* at the n-th level.
- $pCF^{(n)} : \bigcup_{i=1}^n P_{v_i} \times \bigcup_{i=1}^n P_{v_i} \to [0,1]^t$: the DCF, satisfying:
  (1) *Reflexivity*: $pCF^{(n)}(a,a) = 0$,
  (2) *Symmetry*: $pCF^{(n)}(a,b) = pCF^{(n)}(b,a)$.

3. Result: Application of HyperNeutrosophic Sets to Various Sciences

In this section, we explore the application of HyperNeutrosophic Sets across various scientific domains, following the approach outlined in [83]. It is important to note that if HyperNeutrosophic Sets prove applicable to a specific domain, it is reasonable to assume that Hyperfuzzy Sets and Hyperplithogenic Sets could also be utilized in similar contexts. Moreover, for SuperHyperNeutrosophic Sets, it is equally logical to investigate the potential applications of SuperHyperfuzzy Sets and SuperHyperplithogenic Sets within the same or related fields.

3.1. *Neutrosophic Logic*

Logic is the systematic study of reasoning, involving principles and rules to distinguish valid arguments, truth, and consistency [54, 72, 233]. Neutrosophic Logic builds upon classical and fuzzy logic [288] by introducing three degrees—truth, indeterminacy, and falsity—allowing for nuanced reasoning under uncertainty [18, 30, 45, 97, 102, 212, 239, 245]. The concept of a Neutrosophic Set can be viewed as an application of Neutrosophic Logic within the framework of set theory. As evident from previous discussions and references such as [83], it is both natural and necessary to explicitly extend Neutrosophic Logic into HyperNeutrosophic Logic and n-SuperHyperNeutrosophic Logic. While the discussion here centers on HyperNeutrosophic Sets and n-SuperHyperNeutrosophic Sets, similar analyses can also be conducted for Hyperfuzzy Sets, n-SuperHyperfuzzy Sets, Hyperplithogenic Sets, and n-SuperHyperplithogenic Sets.

Definition 3.1 (Neutrosophic Logic)**.** [237] Let p be a proposition. In Neutrosophic Logic, the truth value of p is given by an ordered triple

$$v(p) = (T, I, F) \in [0, 1]^3,$$

where T denotes the degree of truth, I denotes the degree of indeterminacy, and F denotes the degree of falsity. These satisfy the following condition:

$$0 \ \leq \ T + I + F \ \leq \ 3.$$

Unlike many-valued logics that fix $T + F \leq 1$, Neutrosophic Logic allows T, I, F to vary somewhat independently, thereby capturing paradoxical and uncertain statements more flexibly.

Example 3.2 (Neutrosophic Example)**.** Consider a proposition p with

$$v(p) = (0.7,\ 0.2,\ 0.4).$$

Here $T = 0.7, I = 0.2, F = 0.4$, and $0.7 + 0.2 + 0.4 = 1.3 \leq 3$. Thus p can be viewed as mostly true, with moderate falsity and some degree of indeterminacy.

Definition 3.3 (HyperNeutrosophic Logic)**.** Let p be a proposition. In HyperNeutrosophic Logic, the truth value of p is given by a non-empty subset of $[0,1]^3$:

$$v(p) \subseteq [0,1]^3, \quad v(p) \neq \varnothing,$$

where each element $(T, I, F) \in v(p)$ satisfies $0 \leq T + I + F \leq 3$.

Example 3.4 (HyperNeutrosophic Example)**.** Suppose we have two expert opinions about p. One expert assigns $(T, I, F) = (0.7, 0.2, 0.4)$, and another expert assigns $(T, I, F) = (0.4, 0.1, 0.8)$. Then the HyperNeutrosophic valuation can be taken as

$$v(p) = \big\{ (0.7,\, 0.2,\, 0.4),\ (0.4,\, 0.1,\, 0.8) \big\}.$$

This set-based valuation captures multiple sources of uncertain or even conflicting information.

Definition 3.5 (n-SuperHyperNeutrosophic Logic)**.** Let X be a non-empty set and define recursively

$$\tilde{\mathcal{P}}_1(X) = \big\{ A \subseteq X : A \neq \varnothing \big\}, \quad \tilde{\mathcal{P}}_k(X) = \big\{ B \subseteq \tilde{\mathcal{P}}_{k-1}(X) : B \neq \varnothing \big\} \quad (k \geq 2).$$

A *n-SuperHyperNeutrosophic valuation* $v(p)$ is defined to be an element of

$$\tilde{\mathcal{P}}_n\big([0,1]^3\big),$$

i.e. an n-th level nested non-empty subset of the unit cube $[0,1]^3$. At every level, each $(T, I, F) \in [0,1]^3$ must satisfy $0 \leq T + I + F \leq 3$.

Example 3.6 ($n = 2$ SuperHyperNeutrosophic Example)**.** An example of a 2-SuperHyperNeutrosophic valuation $v(p)$ could be

$$v(p) = \Big\{ \{ (0.7,\, 0.2,\, 0.4), (0.4,\, 0.1,\, 0.8) \},\ \{ (0.6,\, 0.3,\, 0.2) \} \Big\}.$$

Here each inner set, such as $\{(0.7, 0.2, 0.4), (0.4, 0.1, 0.8)\}$, is itself a valid HyperNeutrosophic subset of $[0,1]^3$. We then collect those subsets into a larger non-empty set, forming a second-level structure.

Theorem 3.7. *It holds as follows.*

(1) When $n = 1$, an n-SuperHyperNeutrosophic valuation is exactly a HyperNeutrosophic valuation.

(2) If we restrict a HyperNeutrosophic valuation to be a singleton $\{(T, I, F)\}$, it recovers Neutrosophic Logic.

Hence n-SuperHyperNeutrosophic Logic generalizes both HyperNeutrosophic and Neutrosophic Logics.

Proof. (1) By definition, for $n = 1$ we have

$$\tilde{\mathcal{P}}_1([0,1]^3) = \{\, A \subseteq [0,1]^3 : A \neq \varnothing \}.$$

Thus a 1-SuperHyperNeutrosophic valuation $v(p)$ is simply a non-empty subset of $[0,1]^3$, which is precisely the definition of a HyperNeutrosophic valuation.
(2) In Neutrosophic Logic, $v(p)$ is a single triple $(T, I, F) \in [0,1]^3$. If we embed it into HyperNeutrosophic Logic by forming the singleton $\{(T, I, F)\}$, this is clearly a non-empty subset of $[0,1]^3$, thus satisfying the HyperNeutrosophic requirements. Therefore, the singleton case of HyperNeutrosophic valuations coincides with Neutrosophic valuations.

Combining these, we see that n-SuperHyperNeutrosophic Logic (for $n = 1$) equals HyperNeutrosophic Logic, while Neutrosophic Logic is recovered as the singleton case within HyperNeutrosophic sets. For $n > 1$, the framework further generalizes these logics by allowing nested families of HyperNeutrosophic sets. □

3.2. *HyperNeutrosophic Graph Neural Network*

A neural network is a computational model inspired by biological neural systems, designed for tasks such as pattern recognition, data classification, and prediction [8, 13, 23, 159, 274, 282, 283]. Building upon this foundation, a Graph Neural Network (GNN) extends neural networks to graph structures, enabling the modeling of relationships between nodes, edges, and their associated features [58, 142, 187, 211, 227, 231, 272, 278, 295, 299]. Readers may refer to the lecture notes or the introduction for further details(cf. [2,58,74,142,187,211,227,231,284,295]). Building on this concept, Hypergraph Neural Networks (HGNNs) extend traditional Graph Neural Networks (GNNs) by utilizing hyperedges to model higher-order relationships involving multiple nodes simultaneously [35, 79, 122, 128, 141, 267, 276]. Additionally, related concepts, such as the n-SuperHypergraph Neural Network, have also been proposed [86].

Considering these aspects, this paper examines Hyperneutrosophic Graph Neural Networks and Superhyperneutrosophic Graph Neural Networks (cf. [86]). First, several graph concepts addressing various types of uncertainty are briefly introduced below.

Definition 3.8 (Unified Framework for Uncertain Graphs)**.** (cf. [88]) Let $G = (V, E)$ be a classical graph, where V is the set of vertices and E is the set of edges. Depending on the type of graph, each vertex $v \in V$ and edge $e \in E$ is associated with membership values to represent various degrees of truth, indeterminacy, falsity, and other measures of uncertainty.

(1) *Fuzzy Graph* (cf. [26, 103, 107, 186, 196, 217, 278])

- Each vertex $v \in V$ is assigned a membership degree $\sigma(v) \in [0,1]$.
- Each edge $e = (u, v) \in E$ is assigned a membership degree $\mu(u, v) \in [0,1]$.

(2) *Intuitionistic Fuzzy Graph (IFG)* (cf. [7, 138, 270, 297])

- Each vertex $v \in V$ has two values: $\mu_A(v) \in [0,1]$ (degree of membership) and $\nu_A(v) \in [0,1]$ (degree of non-membership), satisfying $\mu_A(v) + \nu_A(v) \leq 1$.
- Each edge $e = (u,v) \in E$ has two values: $\mu_B(u,v) \in [0,1]$ and $\nu_B(u,v) \in [0,1]$, with $\mu_B(u,v) + \nu_B(u,v) \leq 1$.

(3) *Neutrosophic Graph* (cf. [32, 33, 113, 129, 150, 247, 259])

- Each vertex $v \in V$ is associated with a triplet

$$\sigma(v) = (\sigma_T(v), \sigma_I(v), \sigma_F(v))$$

, where

$$\sigma_T(v), \sigma_I(v), \sigma_F(v) \in [0,1]$$

and $\sigma_T(v) + \sigma_I(v) + \sigma_F(v) \leq 3$.

- Each edge $e = (u,v) \in E$ is associated with a triplet $\mu(e) = (\mu_T(e), \mu_I(e), \mu_F(e))$.

(4) *Quadripartitioned Neutrosophic Graph (QNG)* (cf. [131–133, 225, 232])

- Each vertex $v \in V$ is associated with a quadripartitioned neutrosophic membership

$$\sigma(v) = (\sigma_1(v), \sigma_2(v), \sigma_3(v), \sigma_4(v))$$

, where

$$\sigma_1(v), \sigma_2(v), \sigma_3(v), \sigma_4(v) \in [0,1]$$

and

$$\sigma_1(v) + \sigma_2(v) + \sigma_3(v) + \sigma_4(v) \leq 4$$

.

- Each edge $e = (u,v) \in E$ is associated with a quadripartitioned membership

$$\sigma(e) = (\sigma_1(e), \sigma_2(e), \sigma_3(e), \sigma_4(e))$$

, satisfying:

$$\sigma_1(e) \leq \min\{\sigma_1(u), \sigma_1(v)\},$$
$$\sigma_2(e) \leq \min\{\sigma_2(u), \sigma_2(v)\},$$
$$\sigma_3(e) \leq \max\{\sigma_3(u), \sigma_3(v)\},$$
$$\sigma_4(e) \leq \max\{\sigma_4(u), \sigma_4(v)\}.$$

Example 3.9 (Fuzzy Graph)**.** Let $G = (V, E)$, where $V = \{v_1, v_2, v_3\}$ and $E = \{(v_1, v_2), (v_2, v_3)\}$. Each vertex $v \in V$ is assigned a membership degree:

$$\sigma(v_1) = 0.8, \quad \sigma(v_2) = 0.5, \quad \sigma(v_3) = 0.7.$$

Each edge $e \in E$ is assigned a membership degree:

$$\mu(v_1, v_2) = 0.6, \quad \mu(v_2, v_3) = 0.9.$$

This defines a Fuzzy Graph where vertices and edges have varying degrees of membership.

Example 3.10 (Neutrosophic Graph)**.** Let $G = (V, E)$, where $V = \{v_1, v_2, v_3\}$ and $E = \{(v_1, v_2), (v_2, v_3)\}$. Each vertex $v \in V$ is associated with a triplet $\sigma(v) = (\sigma_T(v), \sigma_I(v), \sigma_F(v))$:

$$\sigma(v_1) = (0.7, 0.2, 0.1), \quad \sigma(v_2) = (0.6, 0.3, 0.1), \quad \sigma(v_3) = (0.8, 0.1, 0.1).$$

Each edge $e \in E$ is associated with a triplet $\mu(e) = (\mu_T(e), \mu_I(e), \mu_F(e))$:

$$\mu(v_1, v_2) = (0.5, 0.3, 0.2), \quad \mu(v_2, v_3) = (0.6, 0.2, 0.2).$$

This defines a Neutrosophic Graph with truth, indeterminacy, and falsity values for vertices and edges.

The Neutrosophic Graph Neural Network, along with its extensions, the HyperNeutrosophic Graph Neural Network and the SuperHyperNeutrosophic Graph Neural Network, are introduced below. While the discussion here centers on HyperNeutrosophic Sets and n-SuperHyperNeutrosophic Sets, similar analyses can also be conducted for Hyperfuzzy Sets, n-SuperHyperfuzzy Sets, Hyperplithogenic Sets, and n-SuperHyperplithogenic Sets.

Definition 3.11. In general, feature spaces represent the domains of attributes for vertices and edges, denoted by X_V and X_E, respectively [21,56,151]. Aggregation rules are operations that combine features or information from multiple elements, such as vertices or edges, into a unified representation, denoted as $\mathcal{R}_N$ (cf. [43, 167, 188]). Learnable parameters, denoted as Θ, are adjustable variables (e.g., weights in neural networks) optimized during training to improve model performance (cf. [300]).

Definition 3.12 (Neutrosophic Graph Neural Network (N-GNN) [88])**.** A *Neutrosophic Graph Neural Network (N-GNN)* is a GNN that leverages neutrosophic logic to handle uncertain, indeterminate, and inconsistent data in graph-structured settings. Formally, an N-GNN is an 8-tuple:

$$\text{N-GNN} = \Big(G, X_V, X_E, \mathcal{N}_V, \mathcal{N}_E, \mathcal{R}_N, \mathcal{D}_N, \Theta\Big),$$

where:

(1) $G = (V, E)$ is a graph with vertex set V and edge set E.
(2) X_V and X_E are the feature spaces for vertices and edges, respectively.
(3) $\mathcal{N}_V : X_V \to [0, 1]^3$ and $\mathcal{N}_E : X_E \to [0, 1]^3$ are *neutrosophic fuzzification functions*, mapping features to triples (T, I, F) satisfying $T + I + F \leq 3$.
(4) $\mathcal{R}_N$ is a set of *neutrosophic aggregation rules* specifying how neutrosophic information is combined among vertices and edges.

(5) $\mathcal{D}_N$ is a *neutrosophic defuzzification function* that transforms aggregated neutrosophic values into a crisp or probabilistic output (e.g., a real number or a probability vector [174]).

(6) Θ is the set of learnable parameters (e.g., weights in neural layers or rule parameters).

N-GNN Layer. Given a vertex feature $x_v \in X_V$ for $v \in V$ and an edge feature $x_{uv} \in X_E$ for an edge $(u, v) \in E$, the neutrosophic fuzzification layer outputs:

$$\mathcal{N}_V(x_v) = (\mu_T^v, \mu_I^v, \mu_F^v), \quad \mathcal{N}_E(x_{uv}) = (\mu_T^{uv}, \mu_I^{uv}, \mu_F^{uv}),$$

where each triple fulfills $\mu_T + \mu_I + \mu_F \leq 3$.

Neutrosophic Aggregation. Let $\mathrm{AGG}_N(\cdot)$ be a neutrosophic aggregation operator guided by the rule set $\mathcal{R}_N$. For a vertex v, a typical update rule might be:

$$h_v^{(l+1)} = \sigma\Big(\mathrm{AGG}_N\big(\{\, h_u^{(l)},\, \mathcal{N}_E(x_{uv}) \;\mid\; u \in \mathcal{N}(v)\}\big)\Big),$$

where $h_v^{(l)}$ denotes the hidden representation of vertex v at layer l, $\mathcal{N}(v)$ is the neighborhood of v, and σ is a non-linear activation function [68, 230] (e.g., ReLU). After several layers, the defuzzification step $\mathcal{D}_N$ produces a final crisp or probabilistic output.

Example 3.13 (A Simple N-GNN on a Triangular Graph)**.** *Scenario:* Suppose we have a small graph $G = (V, E)$ with three vertices $V = \{A, B, C\}$ and three edges $E = \{(A, B), (B, C), (C, A)\}$ (cf. [264]). Each vertex and edge has certain uncertain features that we wish to model using neutrosophic logic.

(1) *Vertex Features:*

Let us assume each vertex v has a single feature x_v (e.g., an uncertain sensor reading [77]). We define:

$$x_A = 0.7, \quad x_B = 0.5, \quad x_C = 0.9.$$

Since these sensor readings contain some noise or uncertainty, we convert them into neutrosophic triples (T, I, F) as follows:

$$\mathcal{N}_V(x_A) = (0.6,\, 0.3,\, 0.1), \quad \mathcal{N}_V(x_B) = (0.4,\, 0.2,\, 0.4), \quad \mathcal{N}_V(x_C) = (0.8,\, 0.1,\, 0.1).$$

Each triple must satisfy $T + I + F \leq 3$; here they all sum to $1.0 \leq 3$.

(2) *Edge Features:*

Each edge $(u, v) \in E$ also has a feature x_{uv} (e.g., an uncertain measure of connection strength). For simplicity:

$$x_{AB} = 0.2, \quad x_{BC} = 0.6, \quad x_{CA} = 0.4.$$

Using the neutrosophic fuzzification $\mathcal{N}_E$, suppose:

$$\mathcal{N}_E(x_{AB}) = (0.3,\, 0.4,\, 0.3), \quad \mathcal{N}_E(x_{BC}) = (0.5,\, 0.3,\, 0.2), \quad \mathcal{N}_E(x_{CA}) = (0.4,\, 0.2,\, 0.4).$$

(3) *Initial Hidden States:*

Assign an initial hidden representation $h_v^{(0)} \in \mathbb{R}^d$ to each vertex (e.g., $d = 2$ dimensions). For instance:

$$h_A^{(0)} = (1.0,\, 0.0), \quad h_B^{(0)} = (0.5,\, 0.5), \quad h_C^{(0)} = (0.0,\, 1.0).$$

(4) *Neutrosophic Aggregation:*

We define a neutrosophic aggregation rule AGG_N that combines:

$$(\mu_T^v, \mu_I^v, \mu_F^v) \quad \text{for } v \in V \quad \text{and} \quad (\mu_T^{uv}, \mu_I^{uv}, \mu_F^{uv}) \quad \text{for } (u, v) \in E$$

using some operator, e.g., neutrosophic min or a product-based approach adapted for (T, I, F).

A simplified update for vertex A at layer 1 might look like:

$$h_A^{(1)} = \sigma\Big(W \cdot \mathrm{AGG}_N\big(\{\, h_B^{(0)},\, \mathcal{N}_E(x_{AB}),\; h_C^{(0)},\, \mathcal{N}_E(x_{CA})\}\big)\Big),$$

where W is a weight matrix, and σ is an activation function (like ReLU).

(5) *Final Defuzzification:*

After 2 or 3 message-passing layers, each $h_v^{(\mathrm{final})}$ might be defuzzified via $\mathcal{D}_N$ to produce a class label (e.g., in a classification setting) or a real-valued score (in a regression setting). For example, one could aggregate the final (T, I, F) into a single confidence measure by $\mu_T - \mu_F$ or other transformations, and then map $h_v^{(\mathrm{final})}$ to a label.

(6) *Interpretation:*

A higher T in (T, I, F) suggests the data is more likely to be "true" or valid. A higher I indicates indeterminacy or lack of clarity. A higher F signals contradictions or false components.

By tracking these three degrees, the N-GNN can learn to handle nodes or edges with uncertain or conflicting information more effectively than a standard GNN.

This small triangular graph example shows how even a beginner can view Neutrosophic Graph Neural Networks in action. Each vertex and edge has a neutrosophic triple representing its uncertain state, and the GNN aggregates these values through specialized neutrosophic rules. The final output offers a robust way to manage uncertainty, indeterminacy, and contradiction in the data.

Definition 3.14 (HyperNeutrosophic Graph Neural Network (HN-GNN))**.** A *HyperNeutrosophic Graph Neural Network (HN-GNN)* generalizes the N-GNN by allowing each vertex or edge to have a *set* of neutrosophic triples, rather than a single triple. Formally, an HN-GNN is a 9-tuple:

$$\text{HN-GNN} \;=\; \Big(G,\; X_V,\; X_E,\; \mathcal{HN}_V,\; \mathcal{HN}_E,\; \mathcal{R}_{HN},\; \mathcal{D}_{HN},\; \mathrm{AGG}_{set},\; \Theta\Big),$$

where:

(1) $G = (V, E)$ is a graph.

(2) X_V, X_E are vertex and edge feature spaces.

(3) $\mathcal{HN}_V : X_V \to \mathcal{P}([0,1]^3)$ and $\mathcal{HN}_E : X_E \to \mathcal{P}([0,1]^3)$ are *hyperneutrosophic fuzzification functions*, mapping each vertex (or edge) to a non-empty subset of $[0,1]^3$. Each element $(T_k, I_k, F_k) \in \mathcal{HN}_V(x_v)$ (or $\mathcal{HN}_E(x_{uv})$) satisfies $T_k + I_k + F_k \leq 3$.

(4) AGG_{set} is a *set-level aggregation operator* that collapses or summarizes each hyperneutrosophic set into either (a) a single representative triple, or (b) a small set of representative triples used in the subsequent GNN computation.

(5) $\mathcal{R}_{HN}$ is the hyperneutrosophic rule set for combining hyperneutrosophic information from neighboring vertices and edges.

(6) $\mathcal{D}_{HN}$ is the hyperneutrosophic defuzzification function, producing a final crisp output from the hyperneutrosophic representations.

(7) Θ is the set of learnable parameters in the model.

HN-GNN Layer. At each layer l, for a vertex v:

$$\mathcal{HN}_V(x_v) \;=\; \Big\{(\mu_T^v(i),\, \mu_I^v(i),\, \mu_F^v(i)) \;\Big|\; i \in \mathcal{I}_v\Big\} \;\subseteq\; [0,1]^3,$$

where $\mathcal{I}_v$ indexes multiple neutrosophic evaluations. An edge (u, v) has:

$$\mathcal{HN}_E(x_{uv}) \;=\; \Big\{(\mu_T^{uv}(j),\, \mu_I^{uv}(j),\, \mu_F^{uv}(j)) \;\Big|\; j \in \mathcal{J}_{uv}\Big\}.$$

We first aggregate each hyperneutrosophic set into a suitable representation (e.g., average or maximum triple), or keep multiple triples for a richer representation. The node update then proceeds similarly to an N-GNN, but with set-based inputs instead of single triples:

$$h_v^{(l+1)} \;=\; \sigma\Big(\mathrm{AGG}_{HN}\big(\{\, h_u^{(l)}, \mathcal{HN}_E(x_{uv}) \mid u \in \mathcal{N}(v)\}\big)\Big).$$

Definition 3.15 (n-SuperHyperNeutrosophic Graph Neural Network (n-SHN-GNN))**.** An *n-SuperHyperNeutrosophic Graph Neural Network (n-SHN-GNN)* is a further generalization of the HN-GNN, in which each vertex or edge is endowed with an *n-SuperHyperNeutrosophic Set* instead of a HyperNeutrosophic Set. Formally, an n-SHN-GNN is a 9-tuple:

$$\text{n-SHN-GNN} \;=\; \Big(G,\; X_V,\; X_E,\; \mathcal{SHN}_V^{(n)},\; \mathcal{SHN}_E^{(n)},\; \mathcal{R}_{SHN}^{(n)},\; \mathcal{D}_{SHN}^{(n)},\; \mathrm{AGG}_n,\; \Theta\Big),$$

where:

(1) $G = (V, E)$ is a graph.

(2) X_V, X_E are vertex and edge feature spaces.

(3) $\mathcal{SHN}_V^{(n)} : X_V \to \tilde{\mathcal{P}}_n([0,1]^3)$ and $\mathcal{SHN}_E^{(n)} : X_E \to \tilde{\mathcal{P}}_n([0,1]^3)$ map each vertex (or edge) to an n-SuperHyperNeutrosophic Set of neutrosophic triples. Concretely, each vertex

(or edge) is associated with an n-th nested family of subsets of $[0,1]^3$. Each triple (T, I, F) must satisfy $T + I + F \leq 3$.

(4) AGG_n is an aggregation operator that *collapses* each n-SuperHyperNeutrosophic Set into a small number of representative neutrosophic triples (e.g., using hierarchical combination rules).

(5) $\mathcal{R}_{SHN}^{(n)}$ is the rule set for combining n-SuperHyperNeutrosophic information among neighbors.

(6) $\mathcal{D}_{SHN}^{(n)}$ is the defuzzification step, producing the final crisp or fuzzy outputs from the aggregated hierarchical sets.

(7) Θ is the set of trainable parameters.

Hierarchical Set Representation. Let $\tilde{\mathcal{P}}_n([0,1]^3)$ denote the n-th nested power set of the neutrosophic cube. For a vertex v:

$$\mathcal{SHN}_V^{(n)}(x_v) \ \in \ \tilde{\mathcal{P}}_n([0,1]^3),$$

which might be represented recursively:

$$\mathcal{SHN}_V^{(1)}(x_v) \ = \ \tilde{\mathcal{P}}\big([0,1]^3\big), \quad \mathcal{SHN}_V^{(2)}(x_v) \ = \ \tilde{\mathcal{P}}\Big(\mathcal{SHN}_V^{(1)}(x_v)\Big), \quad \ldots$$

At each level, sets of sets of neutrosophic triples are nested, capturing multi-level uncertainties or multi-source conflicting information.

Layer Update in an n-SHN-GNN. At layer l, suppose each vertex v has hidden representation $h_v^{(l)}$. To update $h_v^{(l+1)}$, do:

$$h_v^{(l+1)} \ = \ \sigma\Big(\mathrm{AGG}_n\Big(\big\{\mathcal{SHN}_E^{(n)}(x_{uv}),\, h_u^{(l)}\big\}_{u\in\mathcal{N}(v)}\Big)\Big).$$

Here, AGG_n must systematically process the nested hierarchical sets from each edge (u, v) or from the vertex features $\mathcal{SHN}_V^{(n)}(x_v)$. After a user-defined number of layers, $\mathcal{D}_{SHN}^{(n)}$ is applied to produce the final output (e.g., classification scores or regression values).

Key Properties of an n-SHN-GNN:

- *Deep Hierarchical Uncertainty:* The n-th nested sets encode multiple layers of contradictory, uncertain, or aggregated data sources.
- *Flexible Aggregation:* Each level requires a well-defined rule to merge or reduce the hierarchical sets into workable forms for neural computations.
- *Generalization of All Previous Cases:* Setting $n = 0$ or $n = 1$ reduces to classical or hyperneutrosophic graph neural networks, respectively, thus unifying these frameworks under one hierarchy.

Remark 3.16. The above definitions of N-GNN, HN-GNN, and n-SHN-GNN assume typical forward-pass, layer-by-layer neural network operations. Training is done by gradient-based optimization (e.g., backpropagation [120,170,216,279]) on a loss function that measures predictive

performance. The novel aspect is the representation of edges and vertices with (hyper)neutrosophic or n-superhyperneutrosophic sets of membership values, enabling richer modeling of uncertainty and ambiguity in graph-structured data.

Theorem 3.17 (Generalization Property)**.** *An n-SuperHyperNeutrosophic Graph Neural Network (n-SHN-GNN) strictly generalizes both the HyperNeutrosophic Graph Neural Network (HN-GNN) and the Neutrosophic Graph Neural Network (N-GNN). Specifically:*

- *If $n = 1$, the n-SHN-GNN reduces to the HN-GNN.*
- *If $n = 0$, the n-SHN-GNN reduces to the N-GNN.*

Proof. Case $n = 0$: By definition, an n-SuperHyperNeutrosophic Set becomes a single neutrosophic triple $(T, I, F) \in [0, 1]^3$ if $n = 0$. Consequently, every vertex or edge in the (0)-SHN-GNN is associated with a single neutrosophic triple, which matches exactly the data representation in a standard Neutrosophic Graph Neural Network (N-GNN). Hence, a (0)-SHN-GNN is identical to an N-GNN in all respects (membership representation, aggregator design, defuzzification steps, etc.).

Case $n = 1$: If $n = 1$, the membership for each vertex or edge is a nonempty subset of $[0, 1]^3$, i.e. a HyperNeutrosophic Set, rather than an n-th nested structure. Thus, the architecture becomes exactly that of a HyperNeutrosophic Graph Neural Network (HN-GNN), where each vertex/edge can hold multiple neutrosophic triples simultaneously but not nested sets-of-sets. Hence, a (1)-SHN-GNN is isomorphic to an HN-GNN.

Case $n > 1$: In this situation, each vertex or edge is assigned an n-fold nested hyperstructure of neutrosophic triples, providing a strictly richer representation than either HN-GNN ($n = 1$) or N-GNN ($n = 0$). Therefore, n-SHN-GNN ($n > 1$) strictly generalizes both HN-GNN and N-GNN, as it can simulate them by appropriate "flattening" of membership sets or by choosing $n = 0, 1$. □

Notation 1. *For brevity, let*

$$\text{n-SHN-GNN} = \big(G,\ X_V,\ X_E,\ \mathcal{SHN}_V^{(n)},\ \mathcal{SHN}_E^{(n)},\ \mathcal{R}_{SHN}^{(n)},\ \mathcal{D}_{SHN}^{(n)},\ \mathrm{AGG}_n,\ \Theta\big)$$

be our canonical reference model.

Theorem 3.18 (Well-Definedness of Layer Updates)**.** *Let* AGG_n *be an aggregation operator that maps from*

$$\Big(\tilde{\mathcal{P}}_n\big([0,1]^3\big)\Big)^{|\mathcal{N}(v)|+1} \ \to\ \tilde{\mathcal{P}}_n\big([0,1]^3\big) \quad \text{or} \quad [0,1]^d,$$

for some $d \in \mathbb{N}$. Suppose AGG_n *is* closed *under the domain of membership sets and preserves the condition $T + I + F \le 3$. Then each layer update in an n-SHN-GNN is well-defined:*

$$h_v^{(l+1)} = \sigma\Big(\mathrm{AGG}_n\big(\{\mathcal{SHN}_E^{(n)}(x_{uv}),\ h_u^{(l)}\}_{u\in\mathcal{N}(v)}\big)\Big),$$

yields valid $h_v^{(l+1)}$ in the intended codomain (e.g., $[0,1]^d$).

Proof. By assumption, AGG_n takes as input a finite set of objects that are each either:

- Elements of $\tilde{\mathcal{P}}_n([0,1]^3)$, i.e. n-SuperHyperNeutrosophic sets.
- Real vector embeddings $h_u^{(l)}$ from the preceding layer (if the aggregator merges representation vectors directly).

Since AGG_n is assumed to be closed under the domain of membership sets, it produces an output that remains in $\tilde{\mathcal{P}}_n([0,1]^3)$ (or in $[0,1]^d$). Furthermore, each triple (T,I,F) within the aggregator's output is guaranteed to satisfy $T+I+F \leq 3$. Thus, the output is a *well-defined* n-superhyperneutrosophic representation or a standard vector embedding, suitable for subsequent neural network operations or final defuzzification. The activation function σ (e.g. ReLU) preserves the property of valid real vector outputs or set membership constraints, concluding the well-definedness of each layer update. □

Theorem 3.19 (Continuity of the Forward Pass)**.** *Assume each aggregator* AGG_n *and activation function σ in the n-SHN-GNN is continuous. Then, as a function of the input features $\{x_v\}_{v\in V}$ and $\{x_{uv}\}_{(u,v)\in E}$, the final output of the n-SHN-GNN is continuous.*

Proof. Let L denote the number of layers, and write

$$\left\{h_v^{(l)}\right\}_{v\in V} \quad \text{for} \quad l=0,1,\ldots,L.$$

At $l=0$, we have $h_v^{(0)} = \mathrm{Enc}\big(\mathcal{SHN}_V^{(n)}(x_v)\big)$ or a direct embedding of the vertex features, which is continuous by assumption of the encoding function Enc. The aggregator AGG_n is continuous in its arguments, and σ is also continuous. Therefore, each update

$$h_v^{(l+1)} = \sigma\Big(\mathrm{AGG}_n(\{h_u^{(l)}, \mathcal{SHN}_E^{(n)}(x_{uv})\}_{u\in\mathcal{N}(v)})\Big)$$

is a composition of continuous mappings in terms of $\{h_u^{(l)}\}$ and the input sets $\{\mathcal{SHN}_E^{(n)}(x_{uv})\}$. By induction on the layer index l, continuity is preserved at each layer, culminating in a continuous final output $\{h_v^{(L)}\}_{v\in V}$. Hence the entire forward pass from input feature sets $\{x_v, x_{uv}\}$ to the final output $\{h_v^{(L)}\}$ is continuous. □

Theorem 3.20 (Reduction Homomorphism for Layer Mapping)**.** *Let $\rho_{n\to m}$ be a map $\tilde{\mathcal{P}}_n([0,1]^3) \to \tilde{\mathcal{P}}_m([0,1]^3)$ with $m<n$, defined by recursively selecting or aggregating subsets in the nested structure. Suppose each layer aggregator* AGG_n *commutes with $\rho_{n\to m}$. Then the n-SHN-GNN naturally reduces to an m-SHN-GNN.*

Proof. Define the reduction map $\rho_{n\to m}$ such that for each $A \in \tilde{\mathcal{P}}_n([0,1]^3)$, we find a corresponding $B \in \tilde{\mathcal{P}}_m([0,1]^3)$. Concretely, $\rho_{n\to m}$ flattens the nested subsets from level n down to level m by either discarding certain nesting levels or merging them. If each layer aggregator AGG_n satisfies

$$\rho_{n\to m}\Big(\mathrm{AGG}_n(\{A_i\}_{i\in I})\Big) = \mathrm{AGG}_m(\{\rho_{n\to m}(A_i)\}_{i\in I}),$$

then we have commutativity of aggregator and flattening. Hence, after applying $\rho_{n\to m}$ to each vertex/edge membership set at every layer, the system evolves exactly as if it were an m-SHN-GNN. Consequently, the entire forward pass of the n-SHN-GNN, under $\rho_{n\to m}$, produces the same outputs as the m-SHN-GNN using the aggregator AGG_m. This proves that n-SHN-GNN reduces to an m-SHN-GNN under the existence of such a homomorphism $\rho_{n\to m}$. □

Definition 3.21 (Fixed Point)**.** (cf. [111]) A fixed point of a function $f : X \to X$ is an element $x^* \in X$ such that $f(x^*) = x^*$. Fixed points represent states or solutions where the application of the function leaves the element unchanged.

Theorem 3.22 (Existence and Uniqueness of a Fixed Point under Contractive Aggregation)**.** *Assume each vertex update in an n-SHN-GNN is given by a contraction mapping in the space of real embeddings (or suitably metricized set space). Formally, suppose there exists $\lambda \in (0,1)$ such that for all pairs of states $\mathbf{H}, \mathbf{H}' \in \mathcal{X}^{|V|}$,*

$$d\Big(\mathrm{AGG}_n(\mathbf{H}),\ \mathrm{AGG}_n(\mathbf{H}')\Big)\ \le\ \lambda\, d(\mathbf{H}, \mathbf{H}'),$$

where d is a metric on the space of states. Then there exists a unique fixed point $\mathbf{H}^*$ such that

$$\mathbf{H}^* = \mathrm{AGG}_n\big(\mathbf{H}^*\big).$$

Proof. This theorem is a direct application of the Banach Fixed Point Theorem [110, 126, 192] (or Contraction Mapping Principle [31]). The aggregator AGG_n is interpreted as a function on the entire set of node states $\mathbf{H} \in \mathcal{X}^{|V|}$. By assumption, it is a λ-contraction with $\lambda < 1$. Therefore, there exists a unique fixed point $\mathbf{H}^*$ satisfying $\mathbf{H}^* = \mathrm{AGG}_n(\mathbf{H}^*)$. Existence follows from standard contraction mapping arguments, and uniqueness arises because any other fixed point would produce a contradiction to the strict contraction property. □

Definition 3.23 (Universal Approximation)**.** (cf. [118, 202]) The Universal Approximation Theorem states that a sufficiently large neural network with appropriate activation functions can approximate any continuous function to arbitrary accuracy on a compact domain. This property underlies the expressive power of neural networks in learning complex mappings.

Theorem 3.24 (Universal Approximation of n-Nested Uncertainty)**.** *Let $\mathcal{F}$ be a class of target functions that map from $\tilde{\mathcal{P}}_n([0,1]^3)$-structured input to real output, i.e.,*

$$f : \tilde{\mathcal{P}}_n([0,1]^3) \times \cdots \times \tilde{\mathcal{P}}_n([0,1]^3) \to \mathbb{R}.$$

Suppose each aggregator AGG_n can be realized as a universal approximator for functions over $\tilde{\mathcal{P}}_n([0,1]^3)$. Then an n-SHN-GNN with sufficient hidden layer width and depth can approximate any target function $f \in \mathcal{F}$ arbitrarily well.

Proof. This statement extends the universal approximation property of neural networks to the domain of nested uncertain sets $\tilde{\mathcal{P}}_n([0,1]^3)$. The key idea is that the aggregator AGG_n (plus any standard feedforward sub-layers) must have enough expressive capacity to approximate arbitrary continuous mappings of the inputs from $\tilde{\mathcal{P}}_n([0,1]^3)$. Under standard assumptions of neural universal approximation (e.g., multi-layer perceptrons with sufficient width and suitable activation), we can embed or encode each nested set structure into a finite-dimensional space, apply a universal approximator, and decode as necessary. Provided the aggregator supports transformations rich enough (e.g., a deep parametric function), it can approximate any continuous function on the domain $\tilde{\mathcal{P}}_n([0,1]^3)$. This argument follows the usual universal approximation theorem, adapted to an embedding space for the nested sets. Convergence in approximation is then guaranteed by classical results on feedforward networks with continuous activation functions (e.g., sigmoids or ReLU). □

3.3. *Neutrosophic Cognitive Maps*

A Cognitive Map is a directed graph that models concepts (nodes) and their causal relationships (edges), where the edges are assigned weighted influences [19, 203, 223, 280]. Over time, various extensions have been developed, including Fuzzy Cognitive Maps [17, 162, 168, 199, 201, 210], Intuitionistic Fuzzy Cognitive Maps [69, 134, 175, 200], Neutrosophic Cognitive Maps [9, 148, 185, 194, 207], Dynamic Cognitive Maps [37, 180], Hesitant fuzzy Cognitive Maps [49–51], Rough Cognitive Maps [46, 47], Cognitive Hypermaps [92], and Cognitive n-SuperHypermaps [92].

This subsection focuses on the HyperNeutrosophic Cognitive Map and the n-SuperHyperNeutrosophic Cognitive Map. Their definitions, associated theorems, and relevant properties are detailed below. While the discussion here centers on HyperNeutrosophic Sets and n-SuperHyperNeutrosophic Sets, similar analyses can also be conducted for Hyperfuzzy Sets, n-SuperHyperfuzzy Sets, Hyperplithogenic Sets, and n-SuperHyperplithogenic Sets.

Definition 3.25 (Limit Cycle)**.** (cf. [14]) In general, a limit cycle is a closed trajectory in the phase space of a dynamical system such that trajectories starting in its vicinity asymptotically

approach it (stable limit cycle) or diverge from it (unstable limit cycle). It represents periodic behavior of the system.

Definition 3.26 (Neutrosophic Cognitive Map (NCM))**.** [9, 148, 185, 194, 207] *Neutrosophic Cognitive Map (NCM)* is a directed graph $\mathcal{G} = (C, E)$ whose vertices (concepts) are linked by edges (causal relationships) weighted by neutrosophic triples. Specifically:

(1) $C = \{C_1, C_2, \ldots, C_n\}$ is a finite set of n concepts representing variables, events, or processes in a system.

(2) $E \subseteq C \times C$ is the set of directed edges, where each edge (C_i, C_j) indicates a causal influence from C_i to C_j.

(3) Each edge (C_i, C_j) has a neutrosophic weight

$$W_{ij} = (T_{ij}, I_{ij}, F_{ij}), \quad \text{with } T_{ij} + I_{ij} + F_{ij} \leq 1,$$

where

$$T_{ij} \in [0,1] \quad \text{(truth or positive influence)}, \quad I_{ij} \in [0,1] \quad \text{(indeterminacy)},$$

$$F_{ij} \in [0,1] \quad \text{(falsity or negative influence)}.$$

Adjacency Matrix of an NCM. The adjacency matrix W of an NCM is a matrix whose (i, j)-th entry is $W_{ij} = (T_{ij}, I_{ij}, F_{ij})$. Each row-column entry thus encodes the neutrosophic weights from one concept to another.

State Vector. At time t, the system's state is given by a vector

$$A(t) = [\, a_1(t),\, a_2(t), \ldots, a_n(t)\,],$$

where each $a_i(t) \in [0, 1]$ denotes the *activation level* of concept C_i at time t.

State Update Rule. The state evolves in discrete time. Given the adjacency matrix W, the new state $A(t+1)$ is computed by:

$$A(t+1) \;=\; \text{Threshold}\Big(A(t) \cdot W\Big),$$

where

$$\big(A(t) \cdot W\big)_j = \sum_{i=1}^{n} \big(T_{ij} \cdot a_i(t) \;-\; F_{ij} \cdot a_i(t) \;+\; I_{ij} \cdot a_i(t)\big),$$

and Threshold$(\cdot)$ re-scales or normalizes the result to remain within $[0, 1]^n$.

Fixed Point and Limit Cycle. A *fixed point* is a state A^* where $A^* = \text{Threshold}(A^* \cdot W)$. A *limit cycle* is a finite sequence of states $\{\, A(t), A(t+1), \ldots, A(t+k)\}$ that repeats periodically.

Example 3.27 (NCM Illustration)**.** *Scenario:* Suppose we have three concepts related to a simplified economic model:

- C_1: Employment rate
- C_2: Investment level

- C_3: Consumer confidence

Let these concepts form an NCM with the following edges and neutrosophic weights:

$$W_{12} = (0.6,\, 0.2,\, 0.0), \quad W_{23} = (0.4,\, 0.3,\, 0.1), \quad W_{31} = (0.0,\, 0.2,\, 0.5),$$

and assume no other direct influences exist, so any missing edges have weight $(0, 0, 0)$.

Interpretation of Weights:

- $W_{12} = (0.6, 0.2, 0.0)$: If the employment rate (C_1) rises, it *positively* influences investment level (C_2) with a truth degree of 0.6. There's some uncertainty (0.2) about the relationship, and no negative influence.
- $W_{23} = (0.4, 0.3, 0.1)$: Investment level (C_2) tends to increase consumer confidence (C_3) but with moderate uncertainty and a small negative component (e.g., risk of inflation).
- $W_{31} = (0.0, 0.2, 0.5)$: Consumer confidence (C_3) might *negatively* impact employment rate if, for example, overconfidence leads to unstable spending (falsity = 0.5). There's also 0.2 uncertainty in this link.

State Update:

$$A(t) = [\, a_1(t),\, a_2(t),\, a_3(t)\,].$$

Then

$$A(t+1) = \text{Threshold}\Big(A(t) \cdot W\Big) = \text{Threshold}\Big(\big(\, X_1,\ X_2,\ X_3\big)\Big),$$

where, for instance,

$$X_2 = \sum_{i=1}^{3} \big(T_{i2} - F_{i2} + I_{i2}\big)\; a_i(t).$$

If we pick an initial state $A(0) = [0.5,\, 0.3,\, 0.8]$, we iteratively update $A(1)$, $A(2), \ldots$ until the system stabilizes or settles into a cycle.

Why It's Useful:

- *Capturing Uncertainty*: Each weight includes truth, indeterminacy, and falsity—helpful when exact causal strengths are not fully known.
- *Modeling Complex Feedback Loops*: NCMs can capture cyclical influences (e.g., $C_1 \to C_2$ and $C_2 \to C_3$ and possibly $C_3 \to C_1$).
- *Possible Outcomes*: The model might converge to a fixed point (e.g., stable employment-investment-confidence levels) or oscillate if the feedback loops are strong or uncertain.

This example provides an illustration of how to construct and interpret an NCM in a simple economic context.

Definition 3.28 (HyperNeutrosophic Cognitive Map (HNCM))**.** A *HyperNeutrosophic Cognitive Map (HNCM)* is a generalization of a Neutrosophic Cognitive Map, in which each directed

edge is associated with a *set* of neutrosophic weights rather than a single neutrosophic triple. Formally, let

$$\mathcal{G} = (C, E)$$

be a directed graph where:

- $C = \{C_1, C_2, \ldots, C_n\}$ is a finite set of n concepts.
- $E \subseteq C \times C$ is a set of directed edges.

For each edge $(C_i, C_j) \in E$, the associated weight is a *HyperNeutrosophic Set*

$$W_{ij} \ \subseteq \ [0,1]^3,$$

where each element of W_{ij} is a triple $\big(T_k(C_i,C_j), I_k(C_i,C_j), F_k(C_i,C_j)\big)$ satisfying:

$$0 \ \leq \ T_k(C_i,C_j) \ + \ I_k(C_i,C_j) \ + \ F_k(C_i,C_j) \ \leq \ 3.$$

That is, for edge (C_i, C_j),

$$W_{ij} \ = \ \Big\{ \big(T_k(C_i,C_j), I_k(C_i,C_j), F_k(C_i,C_j)\big) \ \big| \ k \in \mathcal{K}_{ij} \Big\} \ \subseteq \ [0,1]^3,$$

where $\mathcal{K}_{ij}$ is an index set representing multiple evaluations or sources of uncertainty for the causal relationship from C_i to C_j.

HNCM Adjacency Representation. The adjacency structure of an HNCM can be recorded in a matrix $\mathbf{W}$ whose (i,j)-th entry is the hyperneutrosophic set W_{ij}. That is,

$$\mathbf{W} = \big[\, W_{ij} \big], \quad W_{ij} \ \subseteq \ [0,1]^3.$$

State Vector and Update Rule. Let $A(t) = \big[a_1(t), a_2(t), \ldots, a_n(t)\big]$ be the state of the concepts at time t, where $a_i(t) \in [0,1]$. An HNCM typically requires an *aggregation operator* to combine the multiple neutrosophic triples in W_{ij}. One common approach is to define a function $\mathrm{Agg} : \mathcal{P}([0,1]^3) \to [0,1]^3$ that aggregates the set of triples in W_{ij} into a single effective triple:

$$\mathrm{Agg}\big(W_{ij}\big) = \big(\overline{T}_{ij}, \overline{I}_{ij}, \overline{F}_{ij}\big),$$

where

$$\overline{T}_{ij} + \overline{I}_{ij} + \overline{F}_{ij} \ \leq \ 3.$$

Once aggregated, the state update rule can be modeled analogously to a standard NCM, for example:

$$\big(A(t+1)\big)_j = \mathrm{Threshold}\Big(\sum_{i=1}^{n} \Big[\overline{T}_{ij}\, a_i(t) \ - \ \overline{F}_{ij}\, a_i(t) \ + \ \overline{I}_{ij}\, a_i(t)\Big]\Big),$$

where Threshold is a normalization (cf. [20, 36]) or clipping function (cf. [44]) ensuring $(A(t+1))_j \in [0,1]$. Alternative aggregation or update rules may be used depending on the application.

Definition 3.29 (n-SuperHyperNeutrosophic Cognitive Map (n-SHNCM))**.** An *n-SuperHyperNeutrosophic Cognitive Map* is a further generalization of a HyperNeutrosophic Cognitive Map, where each directed edge is associated with an *n-SuperHyperNeutrosophic Set* rather than a HyperNeutrosophic Set. Formally, let

$$\mathcal{G} = (C, E)$$

be a directed graph where:

- $C = \{C_1, C_2, \ldots, C_n\}$ is a finite set of n concepts.
- $E \subseteq C \times C$ is a set of directed edges.

For each edge $(C_i, C_j) \in E$, the associated weight is an *n-SuperHyperNeutrosophic Set*

$$W_{ij}^{(n)} : \tilde{\mathcal{P}}_n(\{(C_i, C_j)\}) \ \rightarrow \ \tilde{\mathcal{P}}_n\big([0,1]^3\big),$$

where $\tilde{\mathcal{P}}_n$ denotes the n-th nested family of non-empty subsets, as in the definition of n-SuperHyperNeutrosophic Sets. Concretely, $W_{ij}^{(n)}$ assigns to each $A \in \tilde{\mathcal{P}}_n(\{(C_i, C_j)\})$ a subset of $[0,1]^3$, such that:

$$0 \ \leq \ T + I + F \ \leq \ 3$$

for each triple (T, I, F) in that subset.

Intuitive Interpretation. The n-SuperHyperNeutrosophic Set on each edge (C_i, C_j) encapsulates not just multiple neutrosophic evaluations (k-indexed), but multiple levels (or layers) of hierarchical uncertainty. For instance, the first level might capture direct uncertainty from data, the second level might capture expert disagreements about that data, and so forth, up to n nested levels.

Adjacency Representation. The adjacency structure of an n-SHNCM can be encoded in an *n-level superhyperneutrosophic matrix*, with each entry $W_{ij}^{(n)}$ being an n-SuperHyperNeutrosophic Set that maps nested subsets to subsets of $[0,1]^3$.

State Vector and Update Rule. Let

$$A(t) = [\, a_1(t), a_2(t), \ldots, a_n(t) \,]$$

be the state vector at time t. To apply an update, one must first define a procedure to *collapse* or *aggregate* the n-SuperHyperNeutrosophic Set on each edge into an effective triple or small set of triples suitable for computing a concept's activation. For example, one might define:

$$\mathrm{Agg}_n\big(W_{ij}^{(n)}\big) \ \subseteq \ [0,1]^3$$

as an aggregation operator that extracts the relevant truth, indeterminacy, and falsity values from the hierarchical structure. Then, each concept's next state $(A(t+1))_j$ can be computed

using a generalized version of the NCM or HNCM update mechanism:

$$(A(t+1))_j = \text{Threshold}\Big(\sum_{i=1}^{n}\Big[\overline{T}_{ij}\,a_i(t)\ -\ \overline{F}_{ij}\,a_i(t)\ +\ \overline{I}_{ij}\,a_i(t)\Big]\Big),$$

where $\{\overline{T}_{ij}, \overline{I}_{ij}, \overline{F}_{ij}\} \subseteq \text{Agg}_n\big(W_{ij}^{(n)}\big)$ and Threshold ensures the result remains in $[0,1]$.

Key Properties of an n-SHNCM:

- *Hierarchical Uncertainty:* Multiple nested layers of indeterminacy or conflicting data are represented within each edge.
- *Complex Aggregation:* The user must specify an aggregation operator Agg_n to interpret the hierarchical structure for updating concept states.
- *Generalized Dynamics:* Similar to standard cognitive maps, the system may evolve to fixed points, limit cycles, or exhibit chaotic behavior, but now under deeper multi-level uncertainty.

Theorem 3.30 (Generalization Property of n-SuperHyperNeutrosophic Cognitive Maps)**.** *An* n-SuperHyperNeutrosophic Cognitive Map *(n-SHNCM) strictly generalizes both the HyperNeutrosophic Cognitive Map (HNCM) and the Neutrosophic Cognitive Map (NCM). Concretely:*

(1) *If $n = 0$, the n-SHNCM reduces to a standard Neutrosophic Cognitive Map.*

(2) *If $n = 1$, the n-SHNCM reduces to a HyperNeutrosophic Cognitive Map.*

For $n > 1$, it provides additional nesting of neutrosophic information, thereby generalizing both HNCMs ($n = 1$) and NCMs ($n = 0$).

Proof. We recall the following definitions:

- A *Neutrosophic Cognitive Map* (NCM) assigns, to each directed edge (C_i, C_j), exactly one neutrosophic triple (T_{ij}, I_{ij}, F_{ij}), with $T_{ij} + I_{ij} + F_{ij} \leq 1$.
- A *HyperNeutrosophic Cognitive Map* (HNCM) assigns, to each edge (C_i, C_j), a *set* of neutrosophic triples $W_{ij} \subseteq [0,1]^3$. Each triple in that set must satisfy $T + I + F \leq 3$. Typically, after an *aggregation step*, one obtains an effective triple (or small set of triples) to compute causal influences.
- An *n-SuperHyperNeutrosophic Cognitive Map* (n-SHNCM) assigns, to each edge (C_i, C_j), an *n-SuperHyperNeutrosophic Set* of neutrosophic triples. This set is recursively or hierarchically defined up to n levels, capturing additional layers of uncertainty or contradictory evaluations.

Case $n = 0$: By definition, a 0-SuperHyperNeutrosophic Set for each edge is just a single neutrosophic triple $\big(T_{ij}, I_{ij}, F_{ij}\big)$. Hence each edge (C_i, C_j) holds precisely one triple. This matches exactly the data structure of a standard Neutrosophic Cognitive Map. Therefore, when $n = 0$, the n-SHNCM *coincides* with an NCM.

Case $n = 1$: When $n = 1$, each edge (C_i, C_j) is assigned a *HyperNeutrosophic Set* of neutrosophic triples rather than just one triple. This is precisely the definition of a HyperNeutrosophic Cognitive Map (HNCM). Hence for $n = 1$, the n-SHNCM *reduces* to the HNCM framework.

Case $n > 1$: For $n > 1$, an n-SuperHyperNeutrosophic Set is a *nested*, higher-order generalization of a HyperNeutrosophic Set. One obtains successively deeper layers of neutrosophic evaluations. The resulting map can capture more complex or multi-level contradictory opinions. This structure naturally subsumes the single-level sets of a HNCM ($n = 1$) and the single triple of an NCM ($n = 0$).

Hence, for each integer $n \geq 0$:

$$n = 0 \Rightarrow \text{standard NCM}, \quad n = 1 \Rightarrow \text{HNCM}, \quad n > 1 \Rightarrow \text{strict generalization beyond HNCM.}$$

Thus, an n-SHNCM indeed generalizes both HNCM ($n = 1$) and NCM ($n = 0$). □

Theorem 3.31 (Layer-by-Layer Aggregation Consistency)**.** *Let* Agg_n *be an aggregation operator that collapses each n-SuperHyperNeutrosophic Set $W_{ij}^{(n)}$ into finitely many (or one) representative triple(s). Suppose for each edge (C_i, C_j),*

$$\mathrm{Agg}_n\big(W_{ij}^{(n)}\big) \subseteq [0,1]^3,$$

and for every triple (T, I, F) in the output, we have $T + I + F \leq 3$. If Agg_n *is* layer-wise consistent*, i.e. it respects the nested structure in $\tilde{\mathcal{P}}_n\big([0,1]^3\big)$ at each level, then the final aggregated map is well-defined for an n-SHNCM.*

Statement. Layer-wise consistency means that if $V_{ij}^{(k)} \subset \tilde{\mathcal{P}}_k([0,1]^3)$ is a k-fold subset at level k, and $V_{ij}^{(k+1)}$ extends it at level $(k+1)$, then

$$\mathrm{Agg}_{k+1}(V_{ij}^{(k+1)}) \quad \textit{coincides or refines} \quad \mathrm{Agg}_k(V_{ij}^{(k)}),$$

ensuring no contradictions among nested layers. Under this assumption, the aggregator Agg_n *produces a unique or consistent triple set at the final output, guaranteeing a well-defined n-SHNCM adjacency representation.*

Proof. Consider the nested sets:

$$V^{(1)} \subseteq V^{(2)} \subseteq \cdots \subseteq V^{(n)},$$

where each $V^{(k)} \in \tilde{\mathcal{P}}_k([0,1]^3)$. By hypothesis, Agg_n composes Agg_k layer by layer. Specifically, $\mathrm{Agg}_{k+1}\big(V^{(k+1)}\big)$, restricted to level k, must match $\mathrm{Agg}_k\big(V^{(k)}\big)$. This compositional property ensures that the final aggregator output at level n is independent of the path chosen to aggregate intermediate subsets. Hence, the aggregator is well-defined across all edges (C_i, C_j).

As each triple (T, I, F) satisfies $T + I + F \leq 3$, the neutrosophic constraints remain intact, thereby giving a well-defined final adjacency representation for the n-SHNCM. □

Theorem 3.32 (Fixed Point under Contractive Aggregation)**.** *Let an n-SHNCM have a state update rule*

$$A(t+1) = F_n\big(A(t)\big),$$

where $F_n : [0,1]^n \to [0,1]^n$ is formed by aggregating each n-SuperHyperNeutrosophic edge set $W_{ij}^{(n)}$ and summing influences. Assume there exists a metric $d(\cdot,\cdot)$ on $[0,1]^n$ such that F_n is a strict contraction, i.e. for all $A, B \in [0,1]^n$:

$$d\big(F_n(A),\, F_n(B)\big) \;\leq\; \lambda\, d(A,B), \quad \textit{for some constant } \; 0 < \lambda < 1.$$

Then there exists a unique fixed point $A^* \in [0,1]^n$ such that

$$F_n(A^*) = A^*.$$

Proof. This is an application of the Banach Fixed Point Theorem (also known as the Contraction Mapping Principle). Since F_n is defined on the complete metric space $([0,1]^n, d)$ and satisfies $d(F_n(A), F_n(B)) \leq \lambda\, d(A,B)$ with $\lambda < 1$, there is a unique fixed point A^* satisfying $A^* = F_n(A^*)$. Existence follows by iterative updates from any initial state, and uniqueness follows because a strict contraction cannot admit two distinct fixed points. □

Theorem 3.33 (Boundedness of $A(t)$ for n-SHNCM)**.** *In an n-SHNCM, let $A(t) \in [0,1]^n$ evolve via any aggregator-based update rule. Then for all times $t \geq 0$, we have $A(t) \in [0,1]^n$. In other words, the state remains in the unit hypercube $[0,1]^n$ regardless of the complexity or depth n of the superhyperneutrosophic edges.*

Proof. By definition, each aggregator $\mathrm{Agg}_n\big(W_{ij}^{(n)}\big)$ returns neutrosophic triples (T, I, F) with $T + I + F \leq 3$ and each $T, I, F \in [0,1]$. The update for each concept C_j is typically:

$$(A(t+1))_j = \mathrm{Threshold}\Big(\sum_{i=1}^{n} \big[\overline{T}_{ij}\, a_i(t) - \overline{F}_{ij}\, a_i(t) + \overline{I}_{ij}\, a_i(t)\big]\Big),$$

where Threshold is some normalization or clipping that maps real values into $[0,1]$. Since $a_i(t) \in [0,1]$ and $\overline{T}_{ij}, \overline{F}_{ij}, \overline{I}_{ij} \in [0,1]$, the inner expression is bounded (in, say, $[-n, +n]$). The Threshold function ensures $(A(t+1))_j \in [0,1]$. Therefore, $A(t+1) \in [0,1]^n$, and by induction on t (starting from some $A(0) \in [0,1]^n$), all future states remain in $[0,1]^n$. □

3.4. *Neutrosophic Classifier*

A classifier is a function or algorithm that assigns input data to predefined categories or classes based on learned patterns or rules from training data [12, 28, 234]. A Neutrosophic Classifier leverages neutrosophic logic to address uncertainty by assigning truth, indeterminacy, and falsity degrees for classification tasks [15, 16, 71, 285].

A related concept is the Fuzzy Classifier, which uses fuzzy logic to handle uncertainty by assigning membership degrees to classes, enabling flexible and imprecise decision boundaries [39, 165, 208, 301]. Additionally, the Intuitionistic Fuzzy Classifier is another known approach in this context [154, 265, 273].

This section introduces the HyperNeutrosophic Classifier and the n-SuperHyperNeutrosophic Classifier, which extend the Neutrosophic Classifier framework to higher levels of complexity and abstraction.

Definition 3.34 (Supervised Learning)**.** [40, 55, 59, 119] In general, Supervised Learning is a method in machine learning that involves learning a relationship or mapping between a set of input variables X and an output variable Y based on labeled training data. The goal is to construct a model f, which minimizes the risk and accurately predicts the output for unseen data. The training dataset consists of pairs (x_i, y_i), where $x_i \in X$ and $y_i \in Y$. Using the principle of risk minimization, the model f is optimized to generalize well to new data. Once trained, the model can be applied to infer outputs for new, unlabeled inputs.

Notation 2. *All definitions assume a supervised learning context in which we have:*

$$\textit{Training set } \mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\},$$

where $x_i \in \mathbb{R}^d$ (feature space), and $y_i \in \mathcal{C} = \{c_1, c_2, \ldots, c_K\}$ is the set of class labels.

Definition 3.35 (Neutrosophic Classifier)**.** Let $\mathcal{X} = \mathbb{R}^d$ be the feature space, and let $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$ be a finite set of classes. A *Neutrosophic Classifier* is a function

$$\mathcal{N} : \mathcal{X} \ \to \ [0,1]^3 \times \cdots \times [0,1]^3 \ (K \text{ times}),$$

so that for each input $x \in \mathcal{X}$, the classifier outputs a K-tuple of neutrosophic membership triples:

$$\mathcal{N}(x) \ = \ \big((T_1(x), I_1(x), F_1(x)), \ \ldots, \ (T_K(x), I_K(x), F_K(x))\big),$$

where each triple $(T_j(x), I_j(x), F_j(x)) \in [0,1]^3$ satisfies

$$T_j(x) + I_j(x) + F_j(x) \ \leq \ 3.$$

Intuitively, $T_j(x)$ represents the degree of truth-membership (confidence that x is in class c_j), $I_j(x)$ the degree of indeterminacy, and $F_j(x)$ the degree of falsity.

Classification Decision: Typically, one might apply a *defuzzification* or *neutrosophic decision* function to map the neutrosophic outputs to a crisp label in $\mathcal{C}$. For example:

$$\hat{y}(x) = \arg\max_{1\le j\le K}\Big(T_j(x) - F_j(x)\Big),$$

or any other decision rule that accounts for (T_j, I_j, F_j).

Definition 3.36 (HyperNeutrosophic Classifier)**.** Let $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{C} = \{c_1, \ldots, c_K\}$. A *HyperNeutrosophic Classifier* is a function

$$\mathcal{HN} : \mathcal{X} \;\to\; \big(\mathcal{P}([0,1]^3)\big)^K,$$

where for each $x \in \mathcal{X}$ and for each class c_j,

$$\mathcal{HN}(x)_j \;=\; W_j(x) \;\subseteq\; [0,1]^3,$$

is a *HyperNeutrosophic Set* of possible triples $\{(T_k, I_k, F_k)\}$. Each triple (T_k, I_k, F_k) within $W_j(x)$ must satisfy

$$T_k + I_k + F_k \;\le\; 3.$$

Rather than assigning one neutrosophic triple to class c_j, the HyperNeutrosophic Classifier assigns a *nonempty subset* of $[0,1]^3$. Different elements in this subset might represent multiple experts' opinions, different data sources, or uncertain/conflicting evaluations of the membership degrees for class c_j.

Decision Rule: One commonly aggregates each set $W_j(x)$ into a single representative triple, e.g.

$$\overline{(T,I,F)}_j(x) = \mathrm{Agg}\Big(W_j(x)\Big),$$

then apply a neutrosophic-based decision rule, such as

$$\hat{y}(x) = \arg\max_{1\le j\le K}\big(\overline{T}_j(x) - \overline{F}_j(x)\big).$$

Definition 3.37 (n-SuperHyperNeutrosophic Classifier)**.** Let $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{C} = \{c_1, \ldots, c_K\}$.

An *n-SuperHyperNeutrosophic Classifier* is a function

$$\mathcal{SHN}^{(n)} : \mathcal{X} \;\to\; \Big(\tilde{\mathcal{P}}_n\big([0,1]^3\big)\Big)^K,$$

where $\tilde{\mathcal{P}}_n([0,1]^3)$ denotes the n-th nested family of non-empty subsets of the unit cube $[0,1]^3$. Concretely, for each $x \in \mathcal{X}$ and class c_j,

$$\mathcal{SHN}^{(n)}(x)_j \;=\; \mathcal{W}_j^{(n)}(x),$$

where $\mathcal{W}_j^{(n)}(x) \in \tilde{\mathcal{P}}_n([0,1]^3)$. Each nested membership structure satisfies the constraint

$$\forall (T,I,F) \in \mathcal{W}_j^{(n)}(x) \;:\quad T + I + F \;\le\; 3.$$

At level $n = 0$, $\mathcal{W}_j^{(0)}(x)$ is simply a single triple (T, I, F), matching a standard Neutrosophic Classifier. At level $n = 1$, $\mathcal{W}_j^{(1)}(x) \subseteq [0,1]^3$ is a *HyperNeutrosophic set*, i.e. multiple triples.

For $n \geq 2$, one obtains hierarchically nested sets-of-sets, capturing multi-level or multi-source uncertainty at each depth.

Decision Step: One typically defines an aggregation operator $\mathrm{Agg}_n : \tilde{\mathcal{P}}_n([0,1]^3) \to [0,1]^3$ to collapse each nested membership structure to a single triple. Then the classification decision can be performed via a neutrosophic-based rule on these aggregated triples.

Theorem 3.38 (Reduction Property of n-SuperHyperNeutrosophic Classifier)**.** *An n-SuperHyperNeutrosophic Classifier reduces to a HyperNeutrosophic Classifier if $n = 1$, and reduces to a standard Neutrosophic Classifier if $n = 0$.*

Proof. By definition, $\tilde{\mathcal{P}}_0([0,1]^3)$ is just a single triple (T, I, F). Thus, if $n = 0$, $\mathcal{SHN}^{(0)}$ assigns exactly one triple to each class, i.e. a Neutrosophic Classifier. If $n = 1$, $\tilde{\mathcal{P}}_1([0,1]^3) = \mathcal{P}([0,1]^3)$ is a set of one or more triples in $[0,1]^3$, forming a HyperNeutrosophic set. Hence $\mathcal{SHN}^{(1)}$ becomes a HyperNeutrosophic Classifier. For $n > 1$, each class membership is an n-th nested structure, strictly more general than $n = 1$ or $n = 0$. □

Theorem 3.39 (Well-definedness of $\mathcal{SHN}^{(n)}$)**.** *Let $\mathcal{SHN}^{(n)}$ be an n-SuperHyperNeutrosophic Classifier. Suppose for each class c_j, $\mathcal{W}_j^{(n)}(x) \in \tilde{\mathcal{P}}_n([0,1]^3)$ is generated via a function*

$$\Phi_j^{(n)} : \mathcal{X} \to \tilde{\mathcal{P}}_n([0,1]^3),$$

which respects the condition $T + I + F \leq 3$. Then $\mathcal{SHN}^{(n)}$ is well-defined for all $x \in \mathcal{X}$.

Proof. We must check that each $\mathcal{W}_j^{(n)}(x)$ is indeed in $\tilde{\mathcal{P}}_n([0,1]^3)$. By assumption, $\Phi_j^{(n)}$ constructs an element of the n-th nested power set. Moreover, each triple within the structure satisfies $T+I+F \leq 3$. Since $j \in \{1, \ldots, K\}$ is finite, the classifier output is a K-tuple of valid n-SuperHyperNeutrosophic sets. Thus, $\mathcal{SHN}^{(n)}$ is properly defined on the entire domain $\mathcal{X}$. □

Theorem 3.40 (Continuity of a Parametric $\mathcal{SHN}^{(n)}$)**.** *Suppose $\mathcal{SHN}^{(n)}$ depends on parameters Θ, i.e. each $\mathcal{W}_j^{(n)}(x)$ is generated by a continuous function of (x, Θ) into $\tilde{\mathcal{P}}_n([0,1]^3$. If the aggregator and set-construction steps are all continuous with respect to Θ and x, then $\mathcal{SHN}^{(n)}$ is a continuous map from $\mathcal{X} \times \Theta$ into $\left(\tilde{\mathcal{P}}_n([0,1]^3)\right)^K$.*

Proof. Each step in producing $\mathcal{W}_j^{(n)}(x)$ is assumed continuous with respect to the real parameters Θ and the input x. The nested structure $\tilde{\mathcal{P}}_n([0,1]^3)$ can be embedded or represented in a suitable topological space (e.g. via canonical encodings or representing each level's subsets). Under a standard product topology, continuity follows by composition of continuous maps at each stage. Therefore, $\mathcal{SHN}^{(n)}(x; \Theta)$ is continuous in both arguments. □

Theorem 3.41 (Fixed-Structure Theorem for Classifier Consistency)**.** *Consider a training set* $\{(x_i, y_i)\}$*. Suppose the classifier* $\mathcal{SHN}^{(n)}$ *uses a fixed, non-trainable aggregator* Agg_n *that maps each* $\mathcal{W}_j^{(n)}(x)$ *to a single triple* $\overline{(T,I,F)}_j(x)$*. If* Agg_n *and the mapping from input to* $\mathcal{W}_j^{(n)}(x)$ *remain unchanged during training, then the classification boundaries are determined by the aggregator result. Specifically, if* $\hat{y}(x) = \arg\max_j(\overline{T}_j(x) - \overline{F}_j(x))$*, any parameter updates that do not alter* $\mathcal{W}_j^{(n)}(x)$ *or aggregator logic cannot change the decision boundary.*

Proof. Because the aggregator Agg_n and the membership structure $\mathcal{W}_j^{(n)}(x)$ are assumed fixed, none of the computed triple $\overline{(T,I,F)}_j(x)$ can change. Hence, for all x, the final numeric score $\overline{T}_j(x) - \overline{F}_j(x)$ is invariant. Therefore, the classification boundary (i.e. the set of $x \in \mathcal{X}$ where $\max_j(\overline{T}_j(x) - \overline{F}_j(x))$ is tied or changes among classes) remains the same. Thus, no parameter modifications that do not affect $\mathcal{W}_j^{(n)}$ or aggregator logic can alter the classifier's decision function. □

Theorem 3.42 (Universal Approximation under Suitable Encodings)**.** *Let* $\mathcal{SHN}^{(n)}$ *be a parametric n-SuperHyperNeutrosophic classifier, which encodes each n-SuperHyperNeutrosophic set into a finite-dimensional vector (through an appropriate embedding) and then applies a universal approximator (e.g. a multilayer perceptron). Suppose the embedding and aggregator are sufficiently flexible to represent or approximate any continuous function* $\mathcal{X} \to \left(\tilde{\mathcal{P}}_n([0,1]^3)\right)^K$*. Then, for any continuous target classification function* $f^*(x)$ *(mapping* x *to some ideal membership structure in* $\tilde{\mathcal{P}}_n([0,1]^3)$*), there exists a sequence of parameter settings* Θ_m *such that*

$$\lim_{m\to\infty} \left\| \mathcal{SHN}^{(n)}(x;\Theta_m) - f^*(x) \right\| = 0,$$

uniformly on compact subsets of $\mathcal{X}$*. Hence,* $\mathcal{SHN}^{(n)}$ *is a universal approximator in the domain of nested neutrosophic set-based classification.*

Proof. The universal approximation argument proceeds by noting that each element of $\tilde{\mathcal{P}}_n([0,1]^3)$ can be encoded into a finite (though possibly large) dimensional representation (e.g., by enumerating or sampling from the nested membership sets). A sufficiently expressive neural network or parametric system can approximate any continuous mapping from $\mathcal{X} \to \mathcal{Z}$ for $\mathcal{Z} \subset \mathbb{R}^M$. Composing this neural net with a suitable *decoding* step that reconstructs or interprets the aggregated sets ensures that the entire map can approximate any continuous target function f^*. The details rely on standard universal approximation theorems plus a consistent encoding/decoding scheme for the nested sets. Convergence in the sup norm (or uniform metric) follows from classical results in neural net approximation theory. □

3.5. *Neutrosophic Triplet Group*

A Neutrosophic Triplet is an ordered triple that adheres to specific neutral and anti-properties with respect to a binary operation [10, 221, 222, 257, 259]. Related concepts, such as Neutrosophic Duplets, are also recognized in the literature [149, 179, 296]. This framework has been further extended to encompass the HyperNeutrosophic Triplet and the n-SuperHyperNeutrosophic Triplet.

Definition 3.43 (Neutrosophic Triplet)**.** [257] Let $\big(N, \star\big)$ be a nonempty set N with a binary operation $\star : N \times N \to N$. For an element $a \in N$, a *neutrosophic triplet* is an ordered triple $\big(a, \text{neut}(a), \text{anti}(a)\big)$ such that:

$$a \star \text{neut}(a) \;=\; \text{neut}(a) \star a \;=\; a, \quad \text{and} \quad a \star \text{anti}(a) \;=\; \text{anti}(a) \star a \;=\; \text{neut}(a).$$

The elements a, neut(a), and anti(a) are said to form a neutrosophic triplet. In this context:

- neut(a) is called a *neutral of* a (which replaces or generalizes an identity-like element but only relative to a).
- anti(a) is called an *anti of* a (which replaces or generalizes an inverse-like element but only relative to a).

Definition 3.44 (Neutrosophic Triplet Group)**.** A *Neutrosophic Triplet Group (NTG)* is a pair $\big(N, \star\big)$ such that:

(1) *Closure:* For any $x, y \in N$, we have $x \star y \in N$.
(2) *Associativity:* For all $x, y, z \in N$, $(x \star y) \star z = x \star (y \star z)$.
(3) *Existence of Neutrosophic Triplets:* For every $a \in N$, there exist $\text{neut}(a), \text{anti}(a) \in N$ such that

$$a \star \text{neut}(a) = \text{neut}(a) \star a = a, \quad a \star \text{anti}(a) = \text{anti}(a) \star a = \text{neut}(a).$$

We emphasize that neut(a) replaces the notion of a group identity for the specific element a, and anti(a) replaces the notion of an inverse for a. Unlike a classical group, these neutral and anti elements can vary with a.

Example 3.45. Consider $\big(Z_6, \times_6\big)$, where $\times_6$ is multiplication modulo 6. We observe:

$$2 \times_6 4 \;=\; 8 \equiv 2 \pmod 6, \quad 2 \times_6 2 \;=\; 4 \pmod 6.$$

Hence for $a = 2$:

$$\text{neut}(2) = 4, \quad \text{anti}(2) = 2,$$

satisfying

$$2 \star 4 = 4 \star 2 = 2, \quad 2 \star 2 = 4, \quad 4 \star 4 = 4.$$

Thus $\big(2,4,2\big)$ is a neutrosophic triplet. Checking associativity and closure reveals $\big(Z_6,\times_6\big)$ is not a classical group, yet it has valid neutrosophic triplets for certain elements. If each element admits such triplets, it forms an NTG (modulo verifying all conditions).

Definition 3.46 (Commutative NTG)**.** An NTG $\big(N,\star\big)$ is *commutative* if $x\star y = y\star x$ for all $x,y\in N$.

Definition 3.47 (HyperNeutrosophic Triplet Group)**.** A *HyperNeutrosophic Triplet Group (HNTG)* is a pair $\big(N,\star\big)$ such that:

(1) *Closure & Associativity:* For all $x,y\in N$, $x\star y\in N$, and $\star$ is associative.

(2) *HyperNeutrosophic Triplets:* For each $a\in N$, there is a *set of neutrals* $\mathrm{NeutSet}(a)\subseteq N$ and a *set of antis* $\mathrm{AntiSet}(a)\subseteq N$, such that for every $b\in\mathrm{NeutSet}(a)$ and $c\in\mathrm{AntiSet}(a)$,

$$a\star b = b\star a = a,\quad \text{and}\quad a\star c = c\star a = b.$$

In other words, each element a has multiple possible pairs (b,c) forming hyperneutrosophic triplets (a,b,c).

Remark 3.48. In a HyperNeutrosophic Triplet Group, each a can have infinitely many neutrals $\mathrm{NeutSet}(a)$ and infinitely many antis $\mathrm{AntiSet}(a)$. This is an extension of the single-triple idea where we replace $\mathrm{neut}(a)$ by a set of possibilities and $\mathrm{anti}(a)$ by another set.

Example 3.49. Let $\big(N,\star\big)$ be the same set as in a Neutrosophic Triplet Group example, but suppose for each element a, we define a set of neutral candidates $\mathrm{NeutSet}(a)\subseteq N$ and a set of anti candidates $\mathrm{AntiSet}(a)\subseteq N$. As long as for each $b\in\mathrm{NeutSet}(a)$ and $c\in\mathrm{AntiSet}(a)$ the required conditions hold, we get a valid HNTG.

Definition 3.50 (***n***-SuperHyperNeutrosophic Triplet Group)**.** Let n be a nonnegative integer. An *n-SuperHyperNeutrosophic Triplet Group (n-SHNTG)* is a pair $\big(N,\star\big)$ such that:

(1) *Closure & Associativity:* $\star$ is associative and closed on N.

(2) *Nested Triplets Up to Level n:* For each $a\in N$, we have an n-fold nested family of possible neutrals and antis. More explicitly, at level $k\le n$, we define

$$\mathrm{NeutSet}^{(k)}(a)\ \subseteq\ N,\quad \mathrm{AntiSet}^{(k)}(a)\ \subseteq\ N.$$

At each level k, for every pair (b,c) with $b\in\mathrm{NeutSet}^{(k)}(a)$ and $c\in\mathrm{AntiSet}^{(k)}(a)$, the triple (a,b,c) satisfies

$$a\star b = b\star a = a,\quad a\star c = c\star a = b.$$

Additionally, the levels are *nested* in the sense that $\mathrm{NeutSet}^{(k+1)}(a)$ refines or extends $\mathrm{NeutSet}^{(k)}(a)$ and similarly for $\mathrm{AntiSet}^{(k)}(a)$.

Theorem 3.51 (Generalization Property)**.** *Every n-SHNTG with $n = 0$ reduces to a Neutrosophic Triplet Group (NTG), and every n-SHNTG with $n = 1$ reduces to a HyperNeutrosophic Triplet Group (HNTG).*

Proof. If $n = 0$, we allow no hyper-sets of neutrals or antis—only single elements. Hence the definition collapses exactly to a Neutrosophic Triplet Group: each a has a single neut(a) and single anti(a). If $n = 1$, for each a we define $\text{NeutSet}^{(1)}(a)$, $\text{AntiSet}^{(1)}(a)$ as sets of neutrals and antis, recovering the HNTG definition from Definition 3.47. □

Theorem 3.52 (Reduction Homomorphism)**.** *Let $\rho_{n\to m}$ be a surjective map from the n-SuperHyperNeutrosophic structure to an m-SuperHyperNeutrosophic structure, with $m < n$. Suppose*

$$\rho_{n\to m}(a \star b) = \rho_{n\to m}(a) \mathbin{\star} \rho_{n\to m}(b),$$

and similarly for all nest levels of neutrals and antis. Then $(N, \star)$ at level n reduces to (or homomorphically maps onto) the (m)-SHNTG structure.

Proof. We define $\rho_{n\to m}$ to *flatten* or *select subsets* from the n-level hyperstructures. If $\rho_{n\to m}$ respects the binary operation $\star$ (i.e., is an algebraic homomorphism) and commutes with the nested neutrals and antis at each level, the resulting image is a valid m-SHNTG. Details parallel standard homomorphism arguments in universal algebra but adapted to nested triplets. □

Theorem 3.53 (Existence of Trivial Triplets at Each Level)**.** *In an n-SHNTG, for every idempotent element x (i.e. $x \star x = x$) at any level $k \le n$, the triple (x, x, x) forms a trivial (hyper)neutrosophic triplet (or nested family thereof).*

Proof. If $x \star x = x$, then for each $k \le n$, one can place x in $\text{NeutSet}^{(k)}(x)$ and $\text{AntiSet}^{(k)}(x)$, satisfying $x \star x = x$ and $x \star x = x$. □

Theorem 3.54 (Associativity Preservation)**.** *In a HyperNeutrosophic or n-SHNTG, associativity is a global property that must hold for all elements, not just for those forming a single triplet. Consequently, modifying the set of possible neutrals/antis for certain elements must preserve associativity across the entire structure.*

Proof. Follows from standard arguments in universal algebra: The binary operation $\star$ must be globally associative, i.e. $(x \star y) \star z = x \star (y \star z)$ for every $x, y, z \in N$. Defining or modifying hyperneutrosophic sets $\text{NeutSet}^{(k)}(x)$ or $\text{AntiSet}^{(k)}(x)$ does not remove the requirement that $\star$ remains associative on all of N. □

4. Additional Result: Hyperfuzzy Extension

In this section, we explore extensions based on Hyperfuzzy and Superhyperfuzzy concepts rather than HyperNeutrosophic sets.

4.1. *Neuro-Hyperfuzzy System*

A Neuro-Fuzzy System integrates the learning capabilities of neural networks with the reasoning mechanisms of fuzzy logic, enabling effective decision-making in uncertain environments [41, 42, 78, 125, 136, 156, 158]. This hybrid approach leverages the strengths of both neural and fuzzy systems, making it suitable for a wide range of applications. Additionally, a related concept, the Neuro-Neutrosophic System, has been discussed in the literature [89, 271].

In this subsection, we examine the *Neuro-Hyperfuzzy System* and its extension, the *Neuro-Superhyperfuzzy System*. Definitions and relevant details are provided below.

Definition 4.1 (Neuro-Fuzzy System)**.** (cf. [78, 125, 156]) A *Neuro-Fuzzy System* (NFS) is a hybrid intelligent system that integrates fuzzy logic-based reasoning with the learning capabilities of neural networks. Formally, it is represented as a tuple:

$$\mathcal{N} = (\mathbb{X}, \mathbb{Y}, F, R, \mathcal{L}),$$

where:

- $\mathbb{X} \subseteq \mathbb{R}^n$: The input space, where n is the number of input variables.
- $\mathbb{Y} \subseteq \mathbb{R}^m$: The output space, where m is the number of output variables.
- $F = \{f_1, f_2, \ldots, f_p\}$: A set of fuzzy membership functions defined on $\mathbb{X}$.
- $R = \{r_1, r_2, \ldots, r_q\}$: A set of fuzzy rules, each of the form:

$$r_k : \text{IF } \bigwedge_{i=1}^{n} (x_i \text{ is } \mu_{ki}) \text{ THEN } \bigwedge_{j=1}^{m} (y_j \text{ is } \nu_{kj}),$$

  where μ_{ki} and ν_{kj} are fuzzy sets for the inputs and outputs, respectively.
- $\mathcal{L}$: A learning algorithm that adjusts F and R based on training data $T = \{(x_t, y_t)\}_{t=1}^{N}$, where $x_t \in \mathbb{X}$ and $y_t \in \mathbb{Y}$.

Definition 4.2 (System Structure of Neuro-Fuzzy System)**.** The architecture of an NFS can be represented as a multi-layer feedforward network consisting of the following layers:

(1) *Input Layer:* Directly represents the input vector $x = (x_1, x_2, \ldots, x_n)$.

(2) *Fuzzification Layer:* Transforms crisp inputs into fuzzy values using membership functions:

$$\mu_{ki}(x_i) = f_{ki}(x_i), \quad \forall k, i.$$

(3) *Rule Layer:* Computes the firing strength of each fuzzy rule r_k as:

$$\text{Activation: } A_k = \bigwedge_{i=1}^{n} \mu_{ki}(x_i),$$

where $\bigwedge$ is a t-norm (e.g., the minimum operator).

(4) *Aggregation Layer:* Aggregates the outputs of all rules using:

$$\nu(y) = \bigvee_{k=1}^{q} \left(A_k \cdot \nu_k(y)\right),$$

where $\bigvee$ is a t-conorm (e.g., the maximum operator), and $\nu_k(y)$ is the output fuzzy set for rule r_k.

(5) *Defuzzification Layer:* Converts the aggregated fuzzy output into a crisp value:

$$y = \frac{\int_{y \in \mathbb{Y}} y \cdot \nu(y)\, dy}{\int_{y \in \mathbb{Y}} \nu(y)\, dy}.$$

Definition 4.3 (Learning Algorithm)**.** The learning process optimizes the parameters of F and R to minimize a loss function:

$$\mathcal{L}(\Theta) = \frac{1}{N} \sum_{t=1}^{N} \|y_t - \hat{y}(x_t; \Theta)\|^2,$$

where:

- Θ: The set of all parameters, including fuzzy membership function parameters and rule weights.
- $\hat{y}(x_t; \Theta)$: The output of the NFS for input x_t under the current parameter set Θ.

Gradient-based methods or heuristic approaches are employed to adjust Θ iteratively.

Definition 4.4 (Constraints and Interpretability)**.** To ensure interpretability and consistency:

- Fuzzy membership functions μ_{ki} must satisfy overlap constraints (e.g., intersections at membership degree 0.5).
- Rules R should be mutually non-contradictory and logically consistent.

Definition 4.5 (Neuro-Hyperfuzzy System (NHFS))**.** A *Neuro-Hyperfuzzy System* (NHFS) is a hybrid framework integrating a neural network architecture with hyperfuzzy-based membership functions and rules. Formally, we define it as a 5-tuple

$$\mathcal{N}_H = \left(\mathbb{X}, \mathbb{Y}, \tilde{F}, \tilde{R}, \mathcal{L}_H\right),$$

where:

- $\mathbb{X} \subseteq \mathbb{R}^n$: the input space ($n$ real-valued features).
- $\mathbb{Y} \subseteq \mathbb{R}^m$: the output space ($m$ real-valued target variables).

- $\tilde{F} = \{\tilde{f}_1, \ldots, \tilde{f}_p\}$: a set of *hyperfuzzy membership functions*, each

$$\tilde{f}_k : \mathbb{X} \rightarrow \tilde{P}([0,1]),$$

meaning for every input $x \in \mathbb{X}$, $\tilde{f}_k(x) \subseteq [0,1]$ is a (non-empty) set of membership values.

- $\tilde{R} = \{\tilde{r}_1, \ldots, \tilde{r}_q\}$: a set of *hyperfuzzy rules*. Each rule $\tilde{r}_k$ has the form

$$\tilde{r}_k : \text{ IF } \bigwedge_{i=1}^{n} (x_i \text{ is } \tilde{\mu}_{ki}) \text{ THEN } \bigwedge_{j=1}^{m} (y_j \text{ is } \tilde{\nu}_{kj}),$$

where $\tilde{\mu}_{ki}, \tilde{\nu}_{kj}$ are hyperfuzzy membership functions applied to inputs x_i and outputs y_j, respectively.

- $\mathcal{L}_H$: a *learning algorithm* that updates $\tilde{F}$ and $\tilde{R}$ based on training data $\{(x_t, y_t)\}_{t=1}^{N}$, aiming to minimize a specified loss function (e.g., MSE) while accommodating the hyperfuzzy membership framework.

Definition 4.6 (Neuro-n-SuperHyperfuzzy System (NSHFS))**.** Let $\tilde{\mathcal{P}}_n([0,1])$ be the n-SuperHyperfuzzy extension of $[0,1]$. A *Neuro-n-SuperHyperfuzzy System* extends the Neuro-Hyperfuzzy System to n nested levels of hyperfuzzy membership:

$$\mathcal{N}_{SH,n} = \left(\mathbb{X}, \mathbb{Y}, \tilde{F}_n, \tilde{R}_n, \mathcal{L}_{H,n}\right),$$

where:

- $\tilde{F}_n = \{\tilde{f}_{n,1}, \ldots, \tilde{f}_{n,p}\}$, each

$$\tilde{f}_{n,k} : \mathbb{X} \rightarrow \tilde{\mathcal{P}}_n([0,1]),$$

describes membership via n-level nested sets in $[0,1]$.

- $\tilde{R}_n = \{\tilde{r}_{n,1}, \ldots, \tilde{r}_{n,q}\}$, each rule $\tilde{r}_{n,k}$ is:

$$\tilde{r}_{n,k} : \text{ IF } \bigwedge_{i=1}^{n} (x_i \text{ is } \tilde{\mu}_{n,ki}) \text{ THEN } \bigwedge_{j=1}^{m} (y_j \text{ is } \tilde{\nu}_{n,kj}).$$

Here, $\tilde{\mu}_{n,ki}, \tilde{\nu}_{n,kj}$ are n-SuperHyperfuzzy membership functions.

- $\mathcal{L}_{H,n}$ is a *learning algorithm* adapted to handle the n-SuperHyperfuzzy membership structure. It optimizes both $\tilde{F}_n$ and $\tilde{R}_n$ using training data under higher-order uncertainties.

Theorem 4.7 (Universal Approximation for Neuro-Hyperfuzzy Systems)**.** *Let $\mathbb{X} \subseteq \mathbb{R}^n$ be a compact set, and let $f : \mathbb{X} \rightarrow \mathbb{R}^m$ be a continuous function. Then, for every $\varepsilon > 0$, there exists a Neuro-Hyperfuzzy System*

$$\mathcal{N}_H = (\mathbb{X}, \mathbb{Y}, \tilde{F}, \tilde{R}, \mathcal{L}_H)$$

such that for all $x \in \mathbb{X}$,

$$\|\widehat{f}(x) - f(x)\| < \varepsilon,$$

where $\widehat{f}(x)$ is the NHFS output.

Proof. Step 1: Discretization of the Input Space. Since $\mathbb{X}$ is compact in $\mathbb{R}^n$, we can construct a finite covering by hyper-rectangles or grid points $\{x^\ell\}_{\ell=1}^{L}$ such that each point of $\mathbb{X}$ lies within δ-distance of at least one x^ℓ. By the uniform continuity of f on a compact set, there exists $\delta > 0$ ensuring $|f(x) - f(x^\ell)| < \varepsilon/2$ whenever $\|x - x^\ell\| < \delta$.

Step 2: Construction of Hyperfuzzy Membership Functions. For each x^ℓ, define a hyperfuzzy membership function

$$\tilde{f}_\ell : \mathbb{X} \to \tilde{P}([0,1])$$

such that: 1. If x is within δ-distance of x^ℓ, $\tilde{f}_\ell(x)$ includes a subset of membership degrees near 1 (e.g., $[0.8, 1] \subseteq \tilde{f}_\ell(x)$). 2. Otherwise, $\tilde{f}_\ell(x)$ is concentrated near lower membership degrees (e.g., $[0, 0.2]$).

Here, each $\tilde{f}_\ell(x)$ is a *set* of membership degrees rather than a single number, accommodating local variations. Conceptually, this divides $\mathbb{X}$ into overlapping "regions," each associated with a hyperfuzzy membership structure around x^ℓ.

Step 3: Defining Hyperfuzzy Rules. Let $\tilde{R} = \{\tilde{r}_1, \ldots, \tilde{r}_L\}$, where each rule $\tilde{r}_\ell$ is triggered mostly around x^ℓ. We denote an output function $y^\ell \approx f(x^\ell)$ for each ℓ. Then:

$$\tilde{r}_\ell : \quad \text{IF} \bigwedge_{i=1}^{n} \left(x_i \,\text{is}\, \tilde{\mu}_{\ell i}\right) \quad \text{THEN} \quad y \,\text{is}\, \tilde{\nu}_\ell,$$

where $\tilde{\mu}_{\ell i}$ is essentially $\tilde{f}_\ell$ focusing on the i-th dimension, and $\tilde{\nu}_\ell$ is a hyperfuzzy set capturing y^ℓ in $[0, 1]$ for some normalized representation or membership-coded output. The combination of these rules covers the entire input domain.

Step 4: Rule Aggregation and Defuzzification. The system aggregates contributions from each $\tilde{r}_\ell$. When x is near x^ℓ, $\tilde{f}_\ell(x)$ will be high (subset near 1). Thus, rule $\tilde{r}_\ell$ dominates the output, yielding an approximation close to $y^\ell \approx f(x^\ell)$. By uniform continuity, if x is near x^ℓ, $f(x)$ is near $f(x^\ell)$, ensuring

$$\|\widehat{f}(x) - f(x)\| < \varepsilon$$

for appropriate choice of membership boundaries and defuzzification methods.

Step 5: Learning Algorithm. The learning algorithm $\mathcal{L}_H$ refines these memberships and rules to further reduce approximation error. With gradient-based or other optimization methods, the membership sets $\tilde{f}_\ell$ and output sets $\tilde{\nu}_\ell$ converge to a configuration that keeps $\widehat{f}(x)$ within ε of $f(x)$.

Since each step can be made arbitrarily precise by refining the covering and adjusting membership sets, the Neuro-Hyperfuzzy System attains universal approximation. □

Theorem 4.8 (Universal Approximation for Neuro-n-SuperHyperfuzzy Systems)**.** *Let $\mathbb{X} \subseteq \mathbb{R}^n$ be compact, and let $f : \mathbb{X} \to \mathbb{R}^m$ be continuous. Then, for every $\varepsilon > 0$, there exists a Neuro-n-SuperHyperfuzzy System*

$$\mathcal{N}_{SH,n} = \left(\mathbb{X}, \mathbb{Y}, \tilde{F}_n, \tilde{R}_n, \mathcal{L}_{H,n}\right)$$

such that

$$\|\widehat{f}_n(x) - f(x)\| < \varepsilon \quad \textit{for all } x \in \mathbb{X},$$

where $\widehat{f}_n(x)$ is the system output under n-SuperHyperfuzzy membership.

Proof. The construction is similar to that in Theorem 4.7, but each membership function $\tilde{f}_{n,k}$ and rule $\tilde{r}_{n,k}$ uses an n-SuperHyperfuzzy Set, providing nested or higher-level uncertainties.

Step 1: Nested Membership Definition. For each grid point x^ℓ, define

$$\tilde{f}_{n,\ell} : \mathbb{X} \to \tilde{\mathcal{P}}_n([0,1]),$$

so that $\tilde{f}_{n,\ell}(x)$ includes multi-level membership subsets. Each level (1 through n) refines the degree of confidence or uncertainty, allowing flexible overlaps among different "regions" of x-space.

Step 2: n-SuperHyperfuzzy Rules. Construct q rules $\tilde{r}_{n,1}, \ldots, \tilde{r}_{n,q}$, each referencing $\tilde{f}_{n,\ell}$ membership sets. A typical rule might be:

$$\tilde{r}_{n,\ell} : \text{IF } \bigwedge_{i=1}^{n} \left(x_i \text{ is } \tilde{\mu}_{n,\ell i}\right) \quad \text{THEN} \quad y \text{ is } \tilde{\nu}_{n,\ell},$$

where $\tilde{\mu}_{n,\ell i}$ is a membership set in $\tilde{\mathcal{P}}_n([0,1])$ capturing the n-level membership for x_i, and $\tilde{\nu}_{n,\ell}$ similarly captures the output membership.

Step 3: Hierarchical Aggregation. When x is close to x^ℓ, the nested membership $\tilde{f}_{n,\ell}(x)$ becomes strongly activated at various levels, thus $\tilde{r}_{n,\ell}$ dominates the final output. Because we still rely on continuity of f and the refinement principle, we can ensure $\|\widehat{f}_n(x) - f(x)\| < \varepsilon$ by sufficiently dense covering in $\mathbb{X}$ and careful arrangement of nested membership sets.

Step 4: Learning Algorithm for n Levels. The learning process $\mathcal{L}_{H,n}$ can simultaneously tune membership sets at each hierarchy (from level 1 to level n). This does not limit approximation capacity; it merely provides additional structure for capturing uncertain or contradictory sources of information. By adjusting these levels, the system can approximate f within ε.

Hence, the Neuro-n-SuperHyperfuzzy System achieves universal approximation under the same topological and continuity assumptions as in the hyperfuzzy case. □

Remark 4.9. The above theorems emphasize that extending fuzzy membership to hyperfuzzy or n-SuperHyperfuzzy structures does not reduce the ability to approximate continuous functions on compact domains; rather, it enlarges the representational space of uncertainty.

Corollary 4.10. *In both Theorem 4.7 and Theorem 4.8, if the system is allowed to increase the number of rules q and refine membership sets arbitrarily, the approximation error can be made arbitrarily small.*

Proof. This corollary follows directly from the proofs of Theorems 4.7 and 4.8, where the coverings can be made finer and membership sets can be tuned more precisely as $q \to \infty$ or as membership boundaries become sharper. □

4.2. *Hyperfuzzy control*

A control system is a framework managing and regulating processes or devices to achieve desired outputs by adjusting inputs(cf. [3, 24, 73, 104, 115, 228]). The Control System is extended using Fuzzy Sets. Its main feature includes operations such as Fuzzification and Defuzzification. The Fuzzy Control System has been extensively studied in various research fields [53, 124, 130, 161, 173, 178, 182, 184, 205, 275, 277, 298]. The definition is provided below.

Definition 4.11. (cf. [67, 67, 140, 195]) Let $X = \{x_1, x_2, \ldots, x_n\}$ be the universe of discourse for input variables and $Y = \{y_1, y_2, \ldots, y_m\}$ for output variables. A fuzzy control system consists of the following components:

(1) *Fuzzification*: A mapping $F_x : X \to [0, 1]$ that transforms crisp input x_i into a fuzzy set $\mu(x_i)$, where $\mu(x_i) \in [0, 1]$ is the membership degree of x_i.

(2) *Fuzzy Rule Base*: A set of linguistic rules R_k of the form:

$$R_k : \text{If } x_1 \text{ is } A_1^k \text{ and } x_2 \text{ is } A_2^k \text{ and } \ldots \text{ then } y \text{ is } B^k,$$

where A_i^k and B^k are fuzzy sets defined on X and Y, respectively.

(3) *Inference Mechanism*: A function $\Phi : \mathcal{F}(X) \to \mathcal{F}(Y)$ that applies fuzzy logical operators (e.g., Min-Max or Max-Product) to infer the fuzzy output based on the rule base.

(4) *Defuzzification*: A process that converts the fuzzy output $\Phi(\mu(X))$ into a crisp value y^* using a defuzzification method such as:

$$y^* = \frac{\int_{y \in Y} y \cdot \mu(y)\, dy}{\int_{y \in Y} \mu(y)\, dy},$$

where $\mu(y)$ is the membership degree of y in the output fuzzy set.

Definition 4.12 (Hyperfuzzy Control System)**.** Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$ be universes of discourse for inputs and outputs, respectively. A *hyperfuzzy control system* extends the classical fuzzy control system by replacing each fuzzy set with a hyperfuzzy set:

(1) *Hyperfuzzification*: A mapping $\tilde{F}_x : X \to \tilde{P}([0,1])$, so each crisp input x_i is mapped to a *non-empty subset* of $[0,1]$ instead of a single membership degree.

(2) *Hyperfuzzy Rule Base*: A set of hyperfuzzy rules $\tilde{R}_k$ of the form:

$$\tilde{R}_k : \quad \text{If } x_1 \text{ is } \tilde{A}_1^k \text{ and } \ldots \text{ then } y \text{ is } \tilde{B}^k,$$

where each $\tilde{A}_i^k, \tilde{B}^k$ is a hyperfuzzy set on X or Y.

(3) *Hyperfuzzy Inference*: An operator $\tilde{\Phi} : \tilde{\mathcal{F}}(X) \to \tilde{\mathcal{F}}(Y)$ that aggregates hyperfuzzy sets. Various extended t-norms/t-conorms or set-based operations can be used to combine membership *sets* instead of single values.

(4) *Hyperfuzzy Defuzzification*: A method mapping the final hyperfuzzy output $\tilde{\Phi}(\tilde{\mu}(X))$ into a crisp value y^*, potentially by selecting representative degrees from each subset or using bounding strategies:

$$y^* \;=\; \mathrm{Defz}\Big(\tilde{\Phi}\big(\tilde{\mu}(X)\big)\Big),$$

where $\mathrm{Defz}(\cdot)$ is an extended defuzzification operator for hyperfuzzy sets.

Definition 4.13 (n-SuperHyperfuzzy Control System)**.** Let X and Y be universes for inputs and outputs, respectively. An *n-SuperHyperfuzzy Control System* is defined by extending the hyperfuzzy components (Definition 4.12) to n nested levels:

(1) *n-SuperHyperfuzzification*: A mapping

$$\tilde{F}_{x,n} : X \;\to\; \tilde{\mathcal{P}}_n([0,1]),$$

associating each input $x_i \in X$ with an n-superhyperfuzzy set of membership degrees in $[0,1]$.

(2) *n-SuperHyperfuzzy Rule Base*: A set of rules $\tilde{R}_{n,k}$ of the form:

$$\tilde{R}_{n,k} : \quad \text{If } x_1 \text{ is } \tilde{A}_{1,n}^k \text{ and } \ldots \text{ then } y \text{ is } \tilde{B}_n^k,$$

where each $\tilde{A}_{i,n}^k$, $\tilde{B}_n^k$ is an n-superhyperfuzzy set on X or Y.

(3) *n-SuperHyperfuzzy Inference*: An operator $\tilde{\Phi}_n$ that merges the multi-level membership subsets according to extended logic operations.

(4) *n-SuperHyperfuzzy Defuzzification*: A process converting the final multi-level hyperfuzzy output into a crisp control output y^*.

Here we present two illustrative theorems on stability and robustness. They assume a dynamical system model:

$$\dot{x}(t) \;=\; F\big(x(t), u(t)\big),$$

where $x(t)$ is the system state and $u(t)$ is the control input produced by a hyperfuzzy or n-superhyperfuzzy controller.

Definition 4.14 (Robustness)**.** (cf. [163, 287]) Robustness refers to the ability of a system, model, or algorithm to maintain its performance or functionality under perturbations, uncertainties, or adverse conditions in its inputs or environment.

Theorem 4.15 (Stability under Hyperfuzzy Control)**.** *Consider a control system with state-space model $\dot{x}(t) = F\big(x(t), u_{\mathrm{HF}}(t)\big)$, where $u_{\mathrm{HF}}(t)$ is generated by a hyperfuzzy control system. Suppose:*

(1) *$F(\cdot,\cdot)$ is continuous and locally Lipschitz in x for each fixed u_{HF}.*
(2) *There exists a hyperfuzzy rule base ensuring that for each neighborhood of the equilibrium $x^* = 0$, the hyperfuzzy inference produces a control input u_{HF} that reduces a Lyapunov function $V(x)$ (cf. [27, 62, 266, 281]).*

Then $x = 0$ is Lyapunov-stable; i.e., for any $\varepsilon > 0$, there exists $\delta > 0$ such that $\|x(0)\| < \delta$ implies $\|x(t)\| < \varepsilon$ for all $t > 0$.

Proof. Lyapunov Function Construction. By hypothesis, we have a scalar function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ continuous, with $V(0) = 0$ and $V(x) > 0$ for $x \neq 0$. Further, assume V is radially unbounded in a local neighborhood of interest.

Hyperfuzzy Control Influence. The hyperfuzzy rule base ensures that when x is near 0, the membership sets produce a control u_{HF} such that $\dot{V}(x) = \nabla V(x) \cdot F(x, u_{\mathrm{HF}})$ remains non-positive. Since membership sets in $[0, 1]$ are now replaced by $\subseteq [0, 1]$, the control u_{HF} can be chosen consistently from these subsets to maintain $\dot{V}(x) \leq 0$.

Local Stability Conclusion. By standard Lyapunov arguments, if $\dot{V}(x) \leq 0$ in a neighborhood around $x = 0$, $x = 0$ is stable. More precisely, for any $\varepsilon > 0$, choose a region $\Omega_\varepsilon = \{x : V(x) < \gamma(\varepsilon)\}$ that implies $\|x\| < \varepsilon$. The local Lipschitz continuity of F and continuity of V confirm that once $x(0)$ is in Ω_ε, $V(x(t))$ cannot increase, hence $x(t)$ remains in Ω_ε. Thus $x = 0$ is Lyapunov-stable. □

Theorem 4.16 (Robustness under n-SuperHyperfuzzy Control)**.** *Let $\dot{x}(t) = F\big(x(t), u_{n\text{-}SHF}(t)\big)$ be controlled by an n-superhyperfuzzy system (Definition 4.13) with multi-level membership sets. Suppose:*

(1) *$F(x, u)$ is continuous in (x, u) and locally Lipschitz in x for each feasible u.*
(2) *The n-superhyperfuzzy rule base can generate control inputs that counteract bounded external disturbances $d(t)$ up to a known magnitude $D_{\max}$.*

Then the closed-loop system is robustly stable against disturbances of magnitude at most $D_{\max}$, provided the n-superhyperfuzzy sets at each level are appropriately tuned to reduce a Lyapunov function or maintain an invariant set.

Proof. We introduce an augmented system model:

$$\dot{x}(t) = F\big(x(t), u_{n\text{-SHF}}(t)\big) + d(t),$$

with $\|d(t)\| \leq D_{\max}$. By the assumption on the n-superhyperfuzzy rule base, each membership set at level $k = 1, \ldots, n$ provides a "range" of possible control actions. For any x in a certain neighborhood, one can choose an action $u_{n\text{-SHF}}$ from these superhyperfuzzy sets to balance or compensate for $d(t)$ within $\|d(t)\| \leq D_{\max}$.

Let $V(x)$ be a Lyapunov function that decreases under $u_{n\text{-SHF}}$ in the absence of disturbance. Because the multi-level membership offers flexible or nested sets of control values, for every x with $\|x\| \leq R$, there exists $u_{n\text{-SHF}}(x)$ such that $\dot{V}(x) \leq -\alpha(\|x\|)$ if $\|d(t)\| \leq D_{\max}$, for some positive function α. Thus $V(x(t))$ cannot grow unbounded. Consequently, $x(t)$ remains in a bounded region (or converges near 0) even under external disturbance up to $D_{\max}$. This shows robust stability in the sense that the system can handle disturbances within the specified bound. □

Remark 4.17. In Theorem 4.16, the nested membership structure of n-superhyperfuzzy sets allows the control logic to switch or adapt among multiple levels of uncertainty, improving robustness compared to single-level hyperfuzzy or classical fuzzy approaches.

5. Future Work: Further Exploration of HyperUncertain Extensions

This section briefly outlines future directions for this research. We anticipate advancements in extending various concepts using Hyperfuzzy Sets, SuperHyperfuzzy Sets, HyperNeutrosophic Sets, SuperHyperNeutrosophic Sets, Hyperplithogenic Sets, and SuperHyperplithogenic Sets.

Potential areas of extension include:

- *Neutrosophic Queueing Systems* [57, 214, 294].
- *Neutrosophic Geometry* [5, 153, 157].
- *Fuzzy Queueing Systems* [152, 204].
- *Fuzzy Matroids* [108, 109, 112, 169, 191, 224].
- *Fuzzy Topology* [4, 171, 209, 263, 269, 293].
- *Fuzzy Geometry* [48, 197, 218, 219].

We hope this research inspires further exploration and expansion in these domains.

Funding

No external funding was received for this research.

Acknowledgments

We extend our heartfelt thanks to everyone who provided invaluable support in completing this paper; this work would not have been possible without the contributions of all those involved. We also express our deep respect and gratitude to the authors of prior studies cited in our references. Lastly, we are sincerely thankful to our readers for their time and interest.

Data Availability

This paper does not involve any data analysis.

Ethical Approval

This article does not involve any research with human participants or animals.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Mujahid Abbas, Ghulam Murtaza, and Florentin Smarandache. *Basic operations on hypersoft sets and hypersoft point*. Infinite Study, 2020.

[2] Ralph Abboud, Ismail Ilkan Ceylan, Martin Grohe, and Thomas Lukasiewicz. The surprising power of graph neural networks with random node initialization. *arXiv preprint arXiv:2010.01179*, 2020.

[3] Tarek F. Abdelzaher, Yixin Diao, Joseph L. Hellerstein, Chenyang Lu, and Xiaoyun Zhu. Introduction to control theory and its application to computing systems. 2008.

[4] Nasim Abdolmaleki, M. Ahmadi, Hadi Tabatabaee Malazi, and Sebastiano Milardo. Fuzzy topology discovery protocol for sdn-based wireless sensor networks. *Simul. Model. Pract. Theory*, 79:54–68, 2017.

[5] Mohammad Abobala and Ahmad Hatip. An algebraic approach to neutrosophic euclidean geometry. *Neutrosophic Sets and Systems*, 43(1):10, 2021.

[6] Ather Abdulrahman Ageeli. A neutrosophic decision-making methods of the key aspects for supply chain management in international business administrations. *International Journal of Neutrosophic Science*, 23(1):155–167, 2023.

[7] Muhammad Akram and Noura Omair Alshehri. Intuitionistic fuzzy cycles and intuitionistic fuzzy trees. *The Scientific World Journal*, 2014(1):305836, 2014.

[8] Qeethara Al-Shayea. Artificial neural networks in medical diagnosis. *International Journal of Research Publication and Reviews*, 2024.

[9] Salah Hasan Saleh Al-subhi, Iliana Pérez Pupo, Roberto García Vacacela, Pedro Piñero, Pérez, and Maikel Yelandi Leyva Vázquez. A new neutrosophic cognitive map with neutrosophic sets on connections, application in project management. *Neutrosophic Sets and Systems*, 22:63–75, 2018.

[10] Mumtaz Ali, Florentin Smarandache, and Mohsin Khan. Study on the development of neutrosophic triplet ring and neutrosophic triplet field. *Mathematics*, 6(4):46, 2018.

[11] Alireza Aliahmadi and Hamed Nozari. Evaluation of security metrics in aiot and blockchain-based supply chain by neutrosophic decision-making method. *Supply Chain Forum: An International Journal*, 24:31 – 42, 2022.

[12] Keith Allan. Classifiers. *Language*, 53(2):285–311, 1977.

[13] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

[14] Andreas S Andreou, NH Mateou, and George A Zombanakis. Optimization in genetically evolved fuzzy cognitive maps supporting decision-making: the limit cycle case. In *Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications, 2004.*, pages 377–378. IEEE, 2004.

[15] Abdul Quaiyum Ansari, Ranjit Biswas, and Swati Aggarwal. Neutrosophic classifier: An extension of fuzzy classifer. *Applied soft computing*, 13(1):563–573, 2013.

[16] Abdul Quaiyum Ansari, Priyanka Sharma, and Manjari Tripathi. Automatic seizure detection using neutrosophic classifier. *Physical and Engineering Sciences in Medicine*, 43:1019–1028, 2020.

[17] Ioannis D. Apostolopoulos and Peter P. Groumpos. Fuzzy cognitive maps: Their role in explainable artificial intelligence. *Applied Sciences*, 2023.

[18] Charles Ashbacher. *Introduction to Neutrosophic logic*. Infinite Study, 2014.

[19] Robert Axelrod. Structure of decision : the cognitive maps of political elites. 2015.

[20] Jimmy Lei Ba. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[21] Francis Bach. Exploring large feature spaces with hierarchical multiple kernel learning. *Advances in neural information processing systems*, 21, 2008.

[22] Jørgen Bang-Jensen and Gregory Gutin. *Classes of directed graphs*, volume 11. Springer, 2018.

[23] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

[24] Robert N. Bateson. Introduction to control system technology. 1973.

[25] Richard Bellman. *Introduction to matrix analysis*. SIAM, 1997.

[26] Anushree Bhattacharya and Madhumangal Pal. A fuzzy graph theory approach to the facility location problem: A case study in the indian banking system. *Mathematics*, 11(13):2992, 2023.

[27] Franco Blanchini. Nonquadratic lyapunov functions for robust control. *Autom.*, 31:451–461, 1995.

[28] Lashon B. Booker, David E. Goldberg, and John H. Holland. Classifier systems and genetic algorithms. *Artif. Intell.*, 40:235–282, 1989.

[29] Hashem Bordbar, Mohammad Rahim Bordbar, Rajab Ali Borzooei, and Young Bae Jun. N-subalgebras of bck= bci-algebras which are induced from hyperfuzzy structures. *Iranian Journal of Mathematical Sciences and Informatics*, 16(2):179–195, 2021.

[30] Salah Bouzina. Fuzzy logic vs. neutrosophic logic: operations logic. *Neutrosophic Sets and Systems*, 14:29–34, 2016.

[31] Robert M Brooks and Klaus Schmitt. The contraction mapping principle and some applications. *Electronic Journal of Differential Equations*, pages 09–90, 2009.

[32] Said Broumi, Mohamed Talea, Assia Bakali, and Florentin Smarandache. Interval valued neutrosophic graphs. *Critical Review, XII*, 2016:5–33, 2016.

[33] Said Broumi, Mohamed Talea, Assia Bakali, and Florentin Smarandache. Single valued neutrosophic graphs. *Journal of New theory*, (10):86–101, 2016.

[34] Richard Hubert Bruck. *A survey of binary systems*, volume 20. Springer, 1971.

[35] Derun Cai, Moxian Song, Chenxi Sun, Baofeng Zhang, linda Qiao, and Hongyan Li. Hypergraph structure learning for hypergraph neural networks. In *International Joint Conference on Artificial Intelligence*, 2022.

[36] Matteo Carandini and David J Heeger. Normalization as a canonical neural computation. *Nature reviews neuroscience*, 13(1):51–62, 2012.

[37] Joao Paulo Carvalho. On the semantics and the use of fuzzy cognitive maps and dynamic cognitive maps in social sciences. *Fuzzy Sets Syst.*, 214:6–19, 2013.

[38] Y. V. M. Cepeda, M. A. R. Guevara, E. J. J. Mogro, and R. P. Tizano. Impact of irrigation water technification on seven directories of the san juan-patoa river using plithogenic n-superhypergraphs based on environmental indicators in the canton of pujili, 2021. *Neutrosophic Sets and Systems*, 74:46–56, 2024.

[39] Xiaoguang Chang and John H Lilly. Evolutionary design of a fuzzy classifier from data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1894–1906, 2004.

[40] Liang Chen, Paul Bentley, Kensaku Mori, Kazunari Misawa, Michitaka Fujiwara, and Daniel Rueckert. Self-supervised learning for medical image analysis using image context restoration. *Medical image analysis*, 58:101539, 2019.

[41] Wei Chen, Wei Chen, Hamid Reza Pourghasemi, M. Panahi, Aiding Kornejady, Jiale Wang, Xiaoshen Xie, and Shubo Cao. Spatial prediction of landslide susceptibility using an adaptive neuro-fuzzy inference system combined with frequency ratio, generalized additive model, and support vector machine techniques. *Geomorphology*, 297:69–85, 2017.

[42] Wei Chen, M. Panahi, Paraskevas Tsangaratos, Himan Shahabi, Ioanna Ilia, Somayeh Panahi, Shao jun Li, Abolfazl Jaafari, and Baharin Bin Ahmad. Applying population-based evolutionary algorithms and a neuro-fuzzy system for modeling landslide susceptibility. *CATENA*, 2019.

[43] Graciela Chichilnisky. Social aggregation rules and continuity. *The Quarterly Journal of Economics*, 97(2):337–352, 1982.

[44] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

[45] Victor Christianto and Florentin Smarandache. A review of seven applications of neutrosophic logic: In cultural psychology, economics theorizing, conflict resolution, philosophy of science, etc. *J*, 2018.

[46] Zhang Chunying, Liu Lu, Ouyang Dong, and Liang Ruitao. Research of rough cognitive map model. In *Advanced Research on Electronic Commerce, Web Application, and Communication: International Conference, ECWAC 2011, Guangzhou, China, April 16-17, 2011. Proceedings, Part II*, pages 224–229. Springer, 2011.

[47] Zhang Chunying, Liu Lu, Liang Ruitao, and Wang Jing. Rough center mining algorithm of rough cognitive map. *Procedia engineering*, 15:3461–3465, 2011.

[48] Terry D Clark, Jennifer M Larson, John N Mordeson, Joshua D Potter, Mark J Wierman, Terry D Clark, Jennifer M Larson, John N Mordeson, Joshua D Potter, and Mark J Wierman. Fuzzy geometry. *Applying Fuzzy Mathematics to Formal Models in Comparative Politics*, pages 65–80, 2008.

[49] Veysel Çoban and Sezi Çevik Onar. Strategic analysis of solar energy pricing process with hesitant fuzzy cognitive map. *Energy Management?Collective and Computational Intelligence with Theory and Applications*, pages 195–227, 2018.

[50] Veysel Çoban and Sezi Çevik Onar. Analysis of solar energy generation capacity using hesitant fuzzy cognitive maps. *International Journal of Computational Intelligence Systems*, 10(1):1149–1167, 2017.

[51] Veysel Çoban and Sezi Çevik Onar. Modeling renewable energy usage with hesitant fuzzy cognitive map. *Complex & Intelligent Systems*, 3:155–166, 2017.

[52] Mihaela Colhon, Monica Tilea, Ana González-Marcos, Alina Stela Resceanu, Florentin Smarandache, and Fermín Navaridas-Nalda. A neutrosophic decision-making model for determining young people's active engagement. *Int. J. Inf. Technol. Decis. Mak.*, 23:569–598, 2023.

[53] Mario Collotta, Renato Ferrero, Edoardo Giusto, Mohammad Ghazi Vakili, Jacopo Grecuccio, Xiangjie Kong, and Ilsun You. A fuzzy control system for energy-efficient wireless devices in the internet of vehicles. *International Journal of Intelligent Systems*, 36:1595 – 1618, 2021.

[54] Irving M Copi, Carl Cohen, and Kenneth McMahon. *Introduction to logic*. Routledge, 2016.

[55] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008.

[56] Theodoros Damoulas and Mark A Girolami. Combining feature spaces for classification. *Pattern Recognition*, 42(11):2671–2683, 2009.

[57] K Dayana, T Poongodi, and B Vennila. Study on single server finite capacity neutrosophic queueing model. *Computational and Applied Mathematics*, 44(1):1–30, 2025.

[58] Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *AAAI Conference on Artificial Intelligence*, 2021.

[59] Remi Denton, Sam Gross, and Rob Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*, 2016.

[60] Muhammet Deveci, Vladimir Simić, and Ali Ebadi Torkayesh. Remanufacturing facility location for automotive lithium-ion batteries: An integrated neutrosophic decision-making model. *Journal of Cleaner Production*, 317:128438, 2021.

[61] Keith Devlin. *The joy of sets: fundamentals of contemporary set theory*. Springer Science & Business Media, 1994.

[62] Moritz Diehl, Rishi Amrit, and James B. Rawlings. A lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control*, 56:703–707, 2011.

[63] Joseph Diestel and B Faires. On vector measures. *Transactions of the American Mathematical Society*, 198:253–271, 1974.

[64] Reinhard Diestel. Graduate texts in mathematics: Graph theory.

[65] Reinhard Diestel. Graph theory 3rd ed. *Graduate texts in mathematics*, 173(33):12, 2005.

[66] Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.

[67] Dimiter Driankov, Hans Hellendoorn, and Michael Reinfrank. An introduction to fuzzy control (2nd ed.). 1996.

[68] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022.

[69] Mehtap Dursun and Guray Gumus. Intuitionistic fuzzy cognitive map approach for the evaluation of supply chain configuration criteria. *Mathematical Methods in the Applied Sciences*, 43(13):7788–7801, 2020.

[70] Philip Ehrlich. *Real numbers, generalizations of the reals, and theories of continua*, volume 242. Springer Science & Business Media, 2013.

[71] Mohamed Elhoseny, Mahmoud Abdel-salam, and Ibrahim M Elhasnony. Extended fuzzy neutrosophic classifier for accurate intrusion detection and classification. *International Journal of Neutrosophic Science (IJNS)*, 24(4), 2024.

[72] Herbert B. Enderton. A mathematical introduction to logic. 1972.

[73] Virgil Eveleigh and David P. Lindorff. Introduction to control system design. *IEEE Transactions on Systems, Man, and Cybernetics*, 3:529–529, 1973.

[74] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

[75] Ilijas Farah. Approximate homomorphisms. *Combinatorica*, 18:335–348, 1998.

[76] Mehdi Fasanghari and Farzad Habibipour Roudsari. The fuzzy evaluation of e-commerce customer satisfaction. 2008.

[77] Noha S Fayed, Mohammed M Elmogy, Ahmed Atwan, and Eman El-Daydamony. Efficient occupancy detection system based on neutrosophic weighted sensors data fusion. *IEEE Access*, 10:13400–13427, 2022.

[78] Shuang Feng and C.L. Philip Chen. Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification. *IEEE Transactions on Cybernetics*, 50:414–424, 2020.

[79] Yifan Feng, Haoxuan You, Zizhao Zhang, R. Ji, and Yue Gao. Hypergraph neural networks. In *AAAI Conference on Artificial Intelligence*, 2018.

[80] Takaaki Fujita. General plithogenic soft rough graphs and some related graph classes. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, page 437.

[81] Takaaki Fujita. Expanding horizons of plithogenic superhyperstructures: Applications in decision-making, control, and neuro systems. 2024.

[82] Takaaki Fujita. Expanding horizons of plithogenic superhyperstructures: Applications in decision-making, control, and neuro systems. Technical report, Center for Open Science, 2024.

[83] Takaaki Fujita. Exploring concepts of hyperfuzzy, hyperneutrosophic, and hyperplithogenic sets. 2024. DOI: 10.13140/RG.2.2.12216.87045.

[84] Takaaki Fujita. Note for hypersoft filter and fuzzy hypersoft filter. *Multicriteria Algorithms With Applications*, 5:32–51, 2024.

[85] Takaaki Fujita. Short note of supertree-width and n-superhypertree-width. *Neutrosophic Sets and Systems*, 77:54–78, 2024.

[86] Takaaki Fujita. Superhypergraph neural networks and plithogenic graph neural networks: Theoretical foundations. *arXiv preprint arXiv:2412.01176*, 2024.

[87] Takaaki Fujita. Survey of intersection graphs, fuzzy graphs and neutrosophic graphs. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, page 114, 2024.

[88] Takaaki Fujita. Survey of intersection graphs, fuzzy graphs and neutrosophic graphs. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, page 114, 2024.

[89] Takaaki Fujita. A theoretical exploration of hyperconcepts: Hyperfunctions, hyperrandomness, hyperdecision-making, and beyond (including a survey of hyperstructures). 2024.

[90] Takaaki Fujita. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*. Biblio Publishing, 2025.

[91] Takaaki Fujita. Exploration of graph classes and concepts for superhypergraphs and n-th power mathematical structures. 2025.

[92] Takaaki Fujita. A theoretical exploration of hyperconcepts: Hyperfunctions, hyperrandomness, hyperdecision-making, and beyond (including a survey of hyperstructures). *Preprint in Researchgate*, 2025.

[93] Takaaki Fujita and Florentin Smarandache. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Second Volume)*. Biblio Publishing, 2024.

[94] Takaaki Fujita and Florentin Smarandache. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Third Volume)*. Biblio Publishing, 2024.

[95] Takaaki Fujita and Florentin Smarandache. A concise study of some superhypergraph classes. *Neutrosophic Sets and Systems*, 77:548–593, 2024.

[96] Takaaki Fujita and Florentin Smarandache. Fundamental computational problems and algorithms for superhypergraphs. 2024.

[97] Takaaki Fujita and Florentin Smarandache. Introduction to upside-down logic: Its deep relation to neutrosophic logic and applications. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Third Volume)*, 2024.

[98] Takaaki Fujita and Florentin Smarandache. A review of the hierarchy of plithogenic, neutrosophic, and fuzzy graphs: Survey and applications. In *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond (Second Volume)*. Biblio Publishing, 2024.

[99] Takaaki Fujita and Florentin Smarandache. A short note for hypersoft rough graphs. *HyperSoft Set Methods in Engineering*, 3:1–25, 2024.

[100] Takaaki Fujita and Florentin Smarandache. Uncertain automata and uncertain graph grammar. *Neutrosophic Sets and Systems*, 74:128–191, 2024.

[101] Takaaki Fujita and Florentin Smarandache. Local-neutrosophic logic and local-neutrosophic sets: Incorporating locality with applications. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, page 51, 2025.

[102] Takaaki Fujita and Florentin Smarandache. Local-neutrosophic logic and local-neutrosophic sets: Incorporating locality with applications. *Advancing Uncertain Combinatorics through Graphization, Hyperization, and Uncertainization: Fuzzy, Neutrosophic, Soft, Rough, and Beyond*, page 51, 2025.

[103] A Nagoor Gani and K Radha. On regular fuzzy graphs. 2008.

[104] K. C. Garner. Introduction to control system performance measurements. 1968.

[105] Masoud Ghods, Zahra Rostami, and Florentin Smarandache. Introduction to neutrosophic restricted superhypergraphs and neutrosophic restricted superhypertrees and several of their properties. *Neutrosophic Sets and Systems*, 50:480–487, 2022.

[106] Jayanta Ghosh and Tapas Kumar Samanta. Hyperfuzzy sets and hyperfuzzy group. *Int. J. Adv. Sci. Technol*, 41:27–37, 2012.

[107] Puspendu Giri, Somnath Paul, and Bijoy Krishna Debnath. A fuzzy graph theory and matrix approach (fuzzy gtma) to select the best renewable energy alternative in india. *Applied Energy*, 358:122582, 2024.

[108] Roy Goetschel and William Voxman. Fuzzy matroids. 1988.

[109] Roy Goetschel and William Voxman. Fuzzy matroid sums and a greedy algorithm. *Fuzzy Sets and Systems*, 37:189–200, 1990.

[110] M Eshaghi Gordji, M Ramezani, Manuel De La Sen, and Yeol Je Cho. On orthogonal sets and banach fixed point theorem. *Fixed Point Theory*, 18(2):569–578, 2017.

[111] Andrzej Granas, James Dugundji, et al. *Fixed point theory*, volume 14. Springer, 2003.

[112] Fu gui Shi. (l, m)-fuzzy matroids. *Fuzzy Sets Syst.*, 160:2387–2400, 2009.

[113] Muhammad Gulistan, Naveed Yaqoob, Zunaira Rashid, Florentin Smarandache, and Hafiz Abdul Wahab. A study on neutrosophic cubic graphs with real life applications in industries. *Symmetry*, 10(6):203, 2018.

[114] Krystal Guo and Bojan Mohar. Hermitian adjacency matrix of digraphs and mixed graphs. *Journal of Graph Theory*, 85(1):217–248, 2017.

[115] Francis Joseph Hale. Introduction to control system analysis and design. 1973.

[116] Mohammad Hamidi, Florentin Smarandache, and Mohadeseh Taghinezhad. *Decision Making Based on Valued Fuzzy Superhypergraphs*. Infinite Study, 2023.

[117] Mohammad Hamidi and Mohadeseh Taghinezhad. *Application of Superhypergraphs-Based Domination Number in Real World*. Infinite Study, 2023.

[118] Barbara Hammer and Kai Gersmann. A note on the universal approximation capability of support vector machines. *neural processing letters*, 17:43–53, 2003.

[119] Trevor Hastie, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Overview of supervised learning. *The elements of statistical learning: Data mining, inference, and prediction*, pages 9–41, 2009.

[120] Robert Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992.

[121] R. Hema, R. Sudharani, and M. Kavitha. A novel approach on plithogenic interval valued neutrosophic hypersoft sets and its application in decision making. *Indian Journal Of Science And Technology*, 2023.

[122] Nasimeh Heydaribeni, Xinrui Zhan, Ruisi Zhang, Tina Eliassi-Rad, and Farinaz Koushanfar. Hypop: Distributed constrained combinatorial optimization leveraging hypergraph neural networks. *ArXiv*, abs/2311.09375, 2023.

[123] Doeko Homan. Very basic set theory. 2023.

[124] Amirsoheil Honarbari, Sajad Najafi-Shad, Mohsen Saffari Pour, S. S. M. Ajarostaghi, and Ali Hassannia. Mppt improvement for pmsg-based wind turbines using extended kalman filter and fuzzy control system. *Energies*, 2021.

[125] Haoyuan Hong, M. Panahi, Ataollah Shirzadi, Tianwu Ma, Junzhi Liu, A-Xing Zhu, Wei Chen, Ioannis Kougias, and Nerantzis Kazakis. Flood susceptibility assessment in hengfeng area coupling adaptive neuro-fuzzy inference system with genetic algorithm and differential evolution. *The Science of the total environment*, 621:1124–1141, 2018.

[126] WA267432 Horn. Some fixed point theorems for compact maps and flows in banach spaces. *Transactions of the American Mathematical Society*, 149(2):391–404, 1970.

[127] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(1):35–40, 1990.

[128] Chao Hu, Ruishi Yu, Binqi Zeng, Yu Zhan, Ying Fu, Quan Zhang, Rongkai Liu, and Heyuan Shi. Hyperattack: Multi-gradient-guided white-box adversarial structure attack of hypergraph neural networks. *ArXiv*, abs/2302.12407, 2023.

[129] Liangsong Huang, Yu Hu, Yuxia Li, PK Kishore Kumar, Dipak Koley, and Arindam Dey. A study of regular and irregular neutrosophic graphs with real life applications. *Mathematics*, 7(6):551, 2019.

[130] Jiage Huo, Jianghua Zhang, and Felix T. S. Chan. A fuzzy control system for assembly line balancing with a three-state degradation process in the era of industry 4.0. *International Journal of Production Research*, 58:7112 – 7129, 2020.

[131] S Satham Hussain, N Durga, Rahmonlou Hossein, and Ghorai Ganesh. New concepts on quadripartitioned single-valued neutrosophic graph with real-life application. *International Journal of Fuzzy Systems*, 24(3):1515–1529, 2022.

[132] S Satham Hussain, Hossein Rashmonlou, R Jahir Hussain, Sankar Sahoo, Said Broumi, et al. Quadripartitioned neutrosophic graph structures. *Neutrosophic Sets and Systems*, 51(1):17, 2022.

[133] Satham Hussain, Jahir Hussain, Isnaini Rosyida, and Said Broumi. Quadripartitioned neutrosophic soft graphs. In *Handbook of Research on Advances and Applications of Fuzzy Sets and Logic*, pages 771–795. IGI Global, 2022.

[134] Dimitris K Iakovidis and Elpiniki Papageorgiou. Intuitionistic fuzzy cognitive maps for medical decision making. *IEEE Transactions on Information Technology in Biomedicine*, 15(1):100–107, 2010.

[135] Aulia Ishak, R. Ginting, and Wang Wanli. Evaluation of e-commerce services quality using fuzzy ahp and topsis. *IOP Conference Series: Materials Science and Engineering*, 1041, 2021.

[136] Abolfazl Jaafari, Eric K. Zenner, M. Panahi, and Himan Shahabi. Hybrid artificial intelligence models based on a neuro-fuzzy system and metaheuristic optimization algorithms for spatial prediction of wildfire probability. *Agricultural and Forest Meteorology*, 2019.

[137] Muhammad Naveed Jafar, Muhammad Haris Saeed, Kainat Muniba Khan, Faten S. Alamri, and Hamiden A. Wahed Khalifa. Distance and similarity measures using max-min operators of neutrosophic hypersoft sets with application in site selection for solid waste management systems. *IEEE Access*, PP:1–1, 2022.

[138] Chiranjibe Jana, Tapan Senapati, Monoranjan Bhowmik, and Madhumangal Pal. On intuitionistic fuzzy g-subalgebras of g-algebras. *Fuzzy Information and Engineering*, 7(2):195–209, 2015.

[139] Thomas Jech. *Set theory: The third millennium edition, revised and expanded*. Springer, 2003.

[140] David F. Jenkins and Kevin M. Passino. An introduction to nonlinear analysis of fuzzy control systems. *J. Intell. Fuzzy Syst.*, 7:75–103, 1999.

[141] Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. Dynamic hypergraph neural networks. In *International Joint Conference on Artificial Intelligence*, 2019.

[142] Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *ArXiv*, abs/2101.11174, 2021.

[143] Calvin Jongsma. Basic set theory and combinatorics. *Undergraduate Texts in Mathematics*, 2019.

[144] Young Bae Jun, Kul Hur, and Kyoung Ja Lee. Hyperfuzzy subalgebras of bck/bci-algebras. *Annals of Fuzzy Mathematics and Informatics*, 2017.

[145] Young Bae Jun, Min Su Kang, and Seok Zun Song. Several types of bipolar fuzzy hyper bck-ideals in hyper bck-algebras. 2012.

[146] Young Bae Jun, Seon Jeong Kim, and Seok Zun Song. Hyper permeable values and energetic sets in bck/bci-algebras. 2020.

[147] Young Bae Jun, Seok-Zun Song, and Seon Jeong Kim. Distances between hyper structures and length fuzzy ideals of bck/bci-algebras based on hyper structures. *Journal of Intelligent & Fuzzy Systems*, 35(2):2257–2268, 2018.

[148] Dr.W.B.Vasantha Kandasamy and Florentin Smarandache. Fuzzy cognitive maps and neutrosophic cognitive maps. 2003.

[149] Ilanthenral Kandasamy and Florentin Smarandache. Algebraic structure of neutrosophic duplets in neutrosophic rings. 2018.

[150] Vasantha Kandasamy, K Ilanthenral, and Florentin Smarandache. *Neutrosophic graphs: a new dimension to graph theory*. Infinite Study, 2015.

[151] Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International conference on learning representations*, 2020.

[152] Chiang Kao, Chang-Chung Li, and Shih-Pin Chen. Parametric programming to the analysis of fuzzy queues. *Fuzzy sets and systems*, 107(1):93–100, 1999.

[153] Chaitali Kar, Bappa Mondal, and Tapan Kumar Roy. An inventory model under space constraint in neutrosophic environment: a neutrosophic geometric programming approach. *Neutrosophic Sets and Systems: An International Book Series in Information Science and Engineering*, 21(2018):93–109, 2018.

[154] B Kavitha, S Karthikeyan, and P Sheeba Maybell. Emerging intuitionistic fuzzy classifiers for intrusion detection system. *Journal of Advances in Information Technology*, 2(2):99–108, 2011.

[155] M Kaviyarasu, Muhammad Aslam, Farkhanda Afzal, Maha Mohammed Saeed, Arif Mehmood, and Saeed Gul. The connectivity indices concept of neutrosophic graph and their application of computer network, highway system and transport network flow. *Scientific Reports*, 14(1):4891, 2024.

[156] Faezehossadat Khademi, Sayed Mohammadmehdi Jamal, Neela Deshpande, and Shreenivas N. Londhe. Predicting strength of recycled aggregate concrete using artificial neural network, adaptive neuro-fuzzy inference system and multiple linear regression. *International journal of sustainable built environment*, 5:355–369, 2016.

[157] Huda E Khalid, Florentin Smarandache, and Ahmed K Essa. *The basic notions for (over, off, under) neutrosophic geometric programming problems*. Infinite Study, 2018.

[158] Ali Khosravi, Ricardo Nicolau Nassar Koury, Luiz Henrique Jorge Machado, and Juan J. Garcia Pabon. Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system. *Sustainable Energy Technologies and Assessments*, 25:146–160, 2018.

[159] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 2017.

[160] Stephen Cole Kleene. General recursive functions of natural numbers. *Mathematische annalen*, 112(1):727–742, 1936.

[161] Dimitrios Kontogiannis, Dimitrios Bargiotas, and Aspassia Daskalopulu. Fuzzy control system for smart energy management in residential buildings based on environmental data. *Energies*, 2021.

[162] Bart Kosko. Fuzzy cognitive maps. *Int. J. Man Mach. Stud.*, 24:65–75, 1986.

[163] Smita Krishnaswamy, Igor L. Markov, and John P. Hayes. Design for robustness. 2013.

[164] Tarun Kumar and Mukesh Kumar Sharma. Neutrosophic decision-making for allocations in solid transportation problems. *OPSEARCH*, 2024.

[165] Ludmila I Kuncheva. *Fuzzy classifier design*, volume 49. Physica, 2012.

[166] Shulin Lan, Hao Zhang, Ray Y. Zhong, and George Q. Huang. A customer satisfaction evaluation model for logistics services using fuzzy analytic hierarchy process. *Ind. Manag. Data Syst.*, 116:1024–1042, 2016.

[167] Jérôme Lang, Gabriella Pigozzi, Marija Slavkovik, and Leendert Van der Torre. Judgment aggregation rules based on minimization. In *Proceedings of the 13th conference on theoretical aspects of rationality and knowledge*, pages 238–246, 2011.

[168] Maikel Leon. Aggregating procedure for fuzzy cognitive maps. *The International FLAIRS Conference Proceedings*, 2023.

[169] Xiaonan Li. Three-way fuzzy matroids and granular computing. *International Journal of Approximate Reasoning*, 114:44–50, 2019.

[170] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, 2020.

[171] Kimfung Liu, Wenzhong Shi, and Hua Zhang. A fuzzy topology-based maximum likelihood classification. *Isprs Journal of Photogrammetry and Remote Sensing*, 66:103–114, 2011.

[172] Yong Lin Liu, Hee Sik Kim, and J. Neggers. Hyperfuzzy subsets and subgroupoids. *J. Intell. Fuzzy Syst.*, 33:1553–1562, 2017.

[173] Zhi Liu, Ardashir Mohammadzadeh, Hamza Turabieh, Majdi M. Mafarja, Shahab S. Band, and Amir H. Mosavi. A new online learned interval type-3 fuzzy control system for solar energy management systems. *IEEE Access*, 9:10498–10508, 2021.

[174] Yingqi Lu, Pai Zhu, Donglin Wang, and Michel Fattouche. Machine learning techniques with probability vector for cooperative spectrum sensing in cognitive radio networks. In *2016 IEEE wireless communications and networking conference*, pages 1–6. IEEE, 2016.

[175] Chao Luo, Nannan Zhang, and Xingyuan Wang. Time series prediction based on intuitionistic fuzzy cognitive map. *Soft Computing*, 24:6835–6850, 2020.

[176] M MAHARIN. An over view on hyper fuzzy subgroups. *Scholar: National School of Leadership*, 9(1.2), 2020.

[177] Mourad Oqla Massa'deh. On hyper q-fuzzy normal hx subgroup, conjugate and its.

[178] Willians Ribeiro Mendes, Fábio Meneghetti Ugulino de Araújo, Ritaban Dutta, and Derek M. Heeren. Fuzzy control system for variable rate irrigation using remote sensing. *Expert Syst. Appl.*, 124:13–24, 2019.

[179] Hamiyet Merkepci and Katy D. Ahmad. On the conditions of imperfect neutrosophic duplets and imperfect neutrosophic triplets. *Galoitica: Journal of Mathematical Structures and Applications*, 2022.

[180] Yuan Miao, Zhi qiang Liu, Shi Li, and Chee Kheong Siew. Dynamical cognitive network-an extension of fuzzy cognitive map. *Proceedings 11th International Conference on Tools with Artificial Intelligence*, pages 43–46, 1999.

[181] Arunodaya Raj Mishra, Dragan Pamucar, Pratibha Rani, Rajeev Shrivastava, and Ibrahim M. Hezam. Assessing the sustainable energy storage technologies using single-valued neutrosophic decision-making framework with divergence measure. *Expert Syst. Appl.*, 238:121791, 2023.

[182] Bijan Moaveni, Fatemeh Rashidi Fathabadi, and Ali Molavi. Fuzzy control system design for wheel slip prevention and tracking of desired speed profile in electric trains. *Asian Journal of Control*, 24:388 – 400, 2020.

[183] E. J. Mogro, J. R. Molina, G. J. S. Canas, and P. H. Soria. Tree tobacco extract (*Nicotiana glauca*) as a plithogenic bioinsecticide alternative for controlling fruit fly (*Drosophila immigrans*) using *n*-superhypergraphs. *Neutrosophic Sets and Systems*, 74:57–65, 2024.

[184] Ghazal Mohsenian, Sadegh Khalili, Mohammad I. Tradat, Yaman M. Manaserh, Srikanth Rangarajan, Anuroop Desu, Dushyant Thakur, Kourosh Nemati, Kanad Ghose, and Bahgat G. Sammakia. A novel integrated fuzzy control system toward automated local airflow management in data centers. *Control Engineering Practice*, 2021.

[185] Kalyan Mondal, Surapati Pramanik, and Nandalal Ghosh. A study on problems of hijras in west bengal based on neutrosophic cognitive maps. 2014.

[186] John N Mordeson and Sunil Mathew. *Advanced topics in fuzzy graph theory*, volume 375. Springer, 2019.

[187] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

[188] Giuseppe Munda. Choosing aggregation rules for composite indicators. *Social indicators research*, 109(3):337–354, 2012.

[189] Z Nazari and B Mosapour. The entropy of hyperfuzzy sets. *Journal of Dynamical Systems and Geometric Theories*, 16(2):173–185, 2018.

[190] Zohreh Nazari and Batool Mosapour. The entropy of hyperfuzzy sets. *Journal of Dynamical Systems and Geometric Theories*, 16:173 – 185, 2018.

[191] Ladislav A. Novak. On goetschel and voxman fuzzy matroids. *Fuzzy Sets Syst.*, 117:407–412, 2001.

[192] Sandra Oltra and Oscar Valero. Banach's fixed point theorem for partial metric spaces. 2004.

[193] Oluseyi Olurotimi, Amir Dembo, and Thomas Kailath. Neural network weight matrix synthesis using optimal control techniques. In *Neural Information Processing Systems*, 1989.

[194] Rodolfo González Ortega, Marcos David Oviedo Rodríguez, Maikel Yelandi Leyva Vázquez, Jesús Estupiñán, Ricardo, João Alcione Sganderla Figueiredo, and Florentin Smarandache. Pestel analysis based on neutrosophic cognitive maps and neutrosophic numbers for the sinos river basin management. 2019.

[195] Lena Osterhagen. An introduction to fuzzy control. 2016.

[196] Madhumangal Pal, Sovan Samanta, and Ganesh Ghorai. *Modern trends in fuzzy graph theory*. Springer, 2020.

[197] Sankar K. Pal and Ashish Ghosh. Fuzzy geometry in image analysis. *Fuzzy Sets and Systems*, 48:23–40, 1992.

[198] Dragan Pamucar, Morteza Yazdani, Radojko Obradovic, Anil Kumar, and Mercedes Torres-Jiménez. A novel fuzzy hybrid neutrosophic decision-making approach for the resilient supplier selection problem. *International Journal of Intelligent Systems*, 35(12):1934–1986, 2020.

[199] Elpiniki I. Papageorgiou. Review study on fuzzy cognitive maps and their applications during the last decade. *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, pages 828–835, 2011.

[200] Elpiniki I Papageorgiou and Dimitris K Iakovidis. Intuitionistic fuzzy cognitive maps. *IEEE Transactions on Fuzzy Systems*, 21(2):342–354, 2012.

[201] Elpiniki I. Papageorgiou and Jose L. Salmeron. A review of fuzzy cognitive maps research during the last decade. *IEEE Transactions on Fuzzy Systems*, 21:66–79, 2013.

[202] Jooyoung Park and Irwin W Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

[203] Michael Peer, Iva K. Brunec, Nora S. Newcombe, and Russell A. Epstein. Structuring knowledge with cognitive maps and cognitive graphs. *Trends in Cognitive Sciences*, 25:37–54, 2020.

[204] Yannis A Phillis and Runtong Zhang. Fuzzy service control of queueing systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(4):503–517, 1999.

[205] Robert Piotrowski. Supervisory fuzzy control system for biological processes in sequencing wastewater batch reactor. *Urban Water Journal*, 17:325 – 332, 2020.

[206] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[207] Surapati Pramanik, Sourendranath Chackrabarti, Nandalal Ghosh, and Kumar Ashutosh. A study on problems of construction workers in west bengal based on neutrosophic cognitive maps. *International Journal of Innovative Research in Science, Engineering and Technology*, 2:6387–6394, 2014.

[208] Mahardhika Pratama, Witold Pedrycz, and Edwin Lughofer. Evolving ensemble fuzzy classifier. *IEEE Transactions on Fuzzy Systems*, 26(5):2552–2567, 2018.

[209] Pao-Ming Pu and Ying ming Liu. Fuzzy topology. i. neighborhood structure of a fuzzy point and moore-smith convergence. *Journal of Mathematical Analysis and Applications*, 76:571–599, 1980.

[210] Dunwang Qin, Zhen Peng, and Lifeng Wu. Deep attention fuzzy cognitive maps for interpretable multivariate time series prediction. *Knowl. Based Syst.*, 275:110700, 2023.

[211] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[212] Mohammad Naved Qureshi and Mohd Vasim Ahamad. An improved method for image segmentation using k-means clustering with neutrosophic logic. *Procedia computer science*, 132:534–540, 2018.

[213] Atiqe Ur Rahman, Muhammad Haris Saeed, and Hamiden Abd El-Wahed Khalifa. Multi-attribute decision-making based on aggregations and similarity measures of neutrosophic hypersoft sets with possibility setting. *Journal of Experimental & Theoretical Artificial Intelligence*, 36:161 – 186, 2022.

[214] Heba Rashad and Mai Mohamed. Neutrosophic theory and its application in various queueing models: case studies. *Neutrosophic Sets and Systems*, 42:117–135, 2021.

[215] Judith Roitman. *Introduction to modern set theory*, volume 8. John Wiley & Sons, 1990.

[216] Raul Rojas and Raúl Rojas. The backpropagation algorithm. *Neural networks: a systematic introduction*, pages 149–182, 1996.

[217] Azriel Rosenfeld. Fuzzy graphs. In *Fuzzy sets and their applications to cognitive and decision processes*, pages 77–95. Elsevier, 1975.

[218] Azriel Rosenfeld. The fuzzy geometry of image subsets. In *Readings in Fuzzy Sets for Intelligent Systems*, pages 633–639. Elsevier, 1993.

[219] Azriel Rosenfeld. Fuzzy geometry: An updated overview. *Information Sciences*, 110(3-4):127–133, 1998.

[220] Benjamin Rossman. Homomorphism preservation theorems. *J. ACM*, 55:15:1–15:53, 2008.

[221] Memet Şahin and Abdullah Kargın. *Neutrosophic triplet metric topology*. Infinite Study, 2019.

[222] Memet Şahin, Abdullah Kargın, and İsmet Yıldız. Neutrosophic triplet field and neutrosophic triplet vector space based on set valued neutrosophic quadruple number. *Quadruple Neutrosophic Theory And Applications*, 1:52, 2020.

[223] Alexei Samsonovich and Bruce L. McNaughton. Path integration and cognitive mapping in a continuous attractor neural network model. *The Journal of Neuroscience*, 17:5900 – 5920, 1997.

[224] Musavarah Sarwar and Muhammad Akram. New applications of m-polar fuzzy matroids. *Symmetry*, 9:319, 2017.

[225] S Satham Hussain, Durga Nagarajan, Hossein Rashmanlou, and Farshid Mofidnakhaei. Novel supply chain decision making model under m-polar quadripartitioned neutrosophic environment. *Journal of Applied Mathematics and Computing*, pages 1–26, 2024.

[226] P Sathya, Nivetha Martin, and Florentine Smarandache. Plithogenic forest hypersoft sets in plithogenic contradiction based multi-criteria decision making. *Neutrosophic Sets and Systems*, 73:668–693, 2024.

[227] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[228] Klaus Schmidt. Control system design. 2005.

[229] Francina Shalini. Trigonometric similarity measures of pythagorean neutrosophic hypersoft sets. *Neutrosophic Systems with Applications*, 2023.

[230] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.

[231] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *ArXiv*, abs/1811.05868, 2018.

[232] Xiaolong Shi, Saeed Kosari, Hossein Rashmanlou, Said Broumi, and S Satham Hussain. Properties of interval-valued quadripartitioned neutrosophic graphs with real-life application. *Journal of Intelligent & Fuzzy Systems*, 44(5):7683–7697, 2023.

[233] Sajjan G. Shiva. Introduction to logic design. 2018.

[234] Tobias Sing, Oliver Sander, Niko Beerenwinkel, and Thomas Lengauer. Rocr: visualizing classifier performance in r. *Bioinformatics*, 21 20:3940–1, 2005.

[235] F. Smarandache. Introduction to superhyperalgebra and neutrosophic superhyperalgebra. *Journal of Algebraic Hyperstructures and Logical Algebras*, 2022.

[236] Florentin Smarandache. Neutrosophic overset, neutrosophic underset, and neutrosophic offset. similarly for neutrosophic over-/under-/offlogic, probability, and statisticsneutrosophic, pons editions brussels, 170 pages book, 2016.

[237] Florentin Smarandache. A unifying field in logics: Neutrosophic logic. In *Philosophy*, pages 1–141. American Research Press, 1999.

[238] Florentin Smarandache. Neutrosophic set-a generalization of the intuitionistic fuzzy set. *International journal of pure and applied mathematics*, 24(3):287, 2005.

[239] Florentin Smarandache. *A unifying field in logics: neutrosophic logic. Neutrosophy, neutrosophic set, neutrosophic probability: neutrsophic logic. Neutrosophy, neutrosophic set, neutrosophic probability.* Infinite Study, 2005.

[240] Florentin Smarandache. Neutrosophic set–a generalization of the intuitionistic fuzzy set. *Journal of Defense Resources Management (JoDRM)*, 1(1):107–116, 2010.

[241] Florentin Smarandache. Hyperuncertain, superuncertain, and superhyperuncertain sets/logics/probabilities/statistics. *Critical Review*, XIV, 2017.

[242] Florentin Smarandache. Extension of soft set to hypersoft set, and then to plithogenic hypersoft set. *Neutrosophic sets and systems*, 22(1):168–170, 2018.

[243] Florentin Smarandache. *Plithogenic set, an extension of crisp, fuzzy, intuitionistic fuzzy, and neutrosophic sets-revisited*. Infinite study, 2018.

[244] Florentin Smarandache. Plithogeny, plithogenic set, logic, probability, and statistics. *arXiv preprint arXiv:1808.03948*, 2018.

[245] Florentin Smarandache. Extended nonstandard neutrosophic logic, set, and probability based on extended nonstandard analysis. *Symmetry*, 11(4):515, 2019.

[246] Florentin Smarandache. n-superhypergraph and plithogenic n-superhypergraph. *Nidus Idearum*, 7:107–113, 2019.

[247] Florentin Smarandache. *Extension of HyperGraph to n-SuperHyperGraph and to Plithogenic n-SuperHyperGraph, and Extension of HyperAlgebra to n-ary (Classical-/Neutro-/Anti-) HyperAlgebra*. Infinite Study, 2020.

[248] Florentin Smarandache. *NeutroGeometry & AntiGeometry are alternatives and generalizations of the Non-Euclidean Geometries (revisited)*, volume 5. Infinite Study, 2021.

[249] Florentin Smarandache. *Introduction to the n-SuperHyperGraph-the most general form of graph today*. Infinite Study, 2022.

[250] Florentin Smarandache. *Practical applications of IndetermSoft Set and IndetermHyperSoft Set and introduction to TreeSoft Set as an extension of the MultiSoft Set*. Infinite Study, 2022.

[251] Florentin Smarandache. *The SuperHyperFunction and the Neutrosophic SuperHyperFunction (revisited again)*, volume 3. Infinite Study, 2022.

[252] Florentin Smarandache. Decision making based on valued fuzzy superhypergraphs. 2023.

[253] Florentin Smarandache. *SuperHyperFunction, SuperHyperStructure, Neutrosophic SuperHyperFunction and Neutrosophic SuperHyperStructure: Current understanding and future directions*. Infinite Study, 2023.

[254] Florentin Smarandache. Foundation of superhyperstructure & neutrosophic superhyperstructure. *Neutrosophic Sets and Systems*, 63(1):21, 2024.

[255] Florentin Smarandache. Superhyperstructure & neutrosophic superhyperstructure, 2024. Accessed: 2024-12-01.

[256] Florentin Smarandache and Mohamed Abdel-Basset. *Optimization Theory Based on Neutrosophic and Plithogenic Sets*. Academic Press, 2020.

[257] Florentin Smarandache and Mumtaz Ali. Neutrosophic triplet group. *Neural Computing and Applications*, 29(7):595–601, 2018.

[258] Florentin Smarandache and Mumtaz Ali. Neutrosophic triplet group (revisited). *Neutrosophic sets and Systems*, 26(1):2, 2019.

[259] Florentin Smarandache and Mumtaz Ali. Neutrosophic triplet group (revisited). *Neutrosophic sets and Systems*, 26(1):2, 2019.

[260] Florentin Smarandache and NM Gallup. Generalization of the intuitionistic fuzzy set to the neutrosophic set. In *International Conference on Granular Computing*, pages 8–42. Citeseer, 2006.

[261] Florentin Smarandache and AA Salama. Neutrosophic crisp set theory. 2015.

[262] Seok-Zun Song, Seon Jeong Kim, and Young Bae Jun. Hyperfuzzy ideals in bck/bci-algebras. *Mathematics*, 5(4):81, 2017.

[263] Alexander P Šostak. On a fuzzy topological structure. In *Proceedings of the 13th Winter School on Abstract Analysis*, pages 89–103. Circolo Matematico di Palermo, 1985.

[264] Teresa E Steele and Timothy D Weaver. The modified triangular graph: a refined method for comparing mortality profiles in archaeological samples. *Journal of Archaeological Science*, 29(3):317–322, 2002.

[265] Eulalia Szmidt, Janusz Kacprzyk, and Marta Kukier. Intuitionistic fuzzy classifier for imbalanced classes. In *Artificial Intelligence and Soft Computing: 12th International Conference, ICAISC 2013, Zakopane, Poland, June 9-13, 2013, Proceedings, Part I 12*, pages 483–492. Springer, 2013.

[266] Kazuo Tanaka, Tsuyoshi Hori, and Hua O. Wang. A multiple lyapunov function approach to stabilization of fuzzy control systems. *IEEE Trans. Fuzzy Syst.*, 11:582–589, 2003.

[267] Lev Telyatnikov, Maria Sofia Bucarelli, Guillermo Bernardez, Olga Zaghen, Simone Scardapane, and Pietro Lió. Hypergraph neural networks through the lens of message passing: A common perspective to homophily and architecture design. *ArXiv*, abs/2310.07684, 2023.

[268] Ankit Thakkar and Kinjal Chaudhari. Predicting stock trend using an integrated term frequency-inverse document frequency-based feature weight matrix with neural networks. *Appl. Soft Comput.*, 96:106684, 2020.

[269] S. P. Tiwari and Anupam K. Singh. Fuzzy preorder, fuzzy topology and fuzzy transition system. In *Indian Conference on Logic and Its Applications*, 2013.

[270] Vakkas Ulucay and Memet Sahin. Intuitionistic fuzzy soft expert graphs with application. *Uncertainty discourse and applications*, 1(1):1–10, 2024.

[271] TS Umamaheswari and P Sumathi. Enhanced firefly algorithm (efa) based gene selection and adaptive neuro neutrosophic inference system (annis) prediction model for detection of circulating tumor cells (ctcs) in breast cancer analysis. *Cluster Computing*, 22:14035–14047, 2019.

[272] Guillaume Verdon, Trevor McCourt, Enxhell Luzhnica, Vikash Singh, Stefan Leichenauer, and Jack Hidary. Quantum graph neural networks. *arXiv preprint arXiv:1909.12264*, 2019.

[273] MR Vinodh and PJ Arui Leena Rose. Intelligent deep adaptive intuitionistic fuzzy classifier based dyslexia prediction among children at its early stage. *Journal of Electrical Systems*, 20(3s):52–61, 2024.

[274] Jia Wang and Zhenyuan Wang. Using neural networks to determine sugeno measures by statistics. *Neural Networks*, 10:183–195, 1997.

[275] Lina Wang, Binrui Wang, and M. Zhu. Multi-model adaptive fuzzy control system based on switch mechanism in a greenhouse. *Applied Engineering in Agriculture*, 36:549–556, 2020.

[276] Yuxin Wang, Quan Gan, Xipeng Qiu, Xuanjing Huang, and David Paul Wipf. From hypergraph energy functions to hypergraph neural networks. In *International Conference on Machine Learning*, 2023.

[277] Zhanshan Wang, Jian Sun, and Huaguang Zhang. Stability analysis of t-s fuzzy control system with sampled-dropouts based on time-varying lyapunov function method. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50:2566–2577, 2020.

[278] Tong Wei, Junlin Hou, and Rui Feng. Fuzzy graph neural network for few-shot learning. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020.

[279] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[280] Andrew M. Wikenheiser and Geoffrey Schoenbaum. Over the river, through the woods: cognitive maps in the hippocampus and orbitofrontal cortex. *Nature Reviews Neuroscience*, 17:513–523, 2016.

[281] F. Wesley Wilson. The structure of the level surfaces of a lyapunov function. *Journal of Differential Equations*, 3:323–329, 1967.

[282] Jianxin Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5(23):495, 2017.

[283] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[284] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[285] Ibrahim Yasser, Abeer Twakol, AA Abd El-Khalek, Ahmed Samrah, and AA Salama. Covid-x: novel health-fog framework based on neutrosophic classifier for confrontation covid-19. *Neutrosophic Sets and Systems*, 35:1–21, 2020.

[286] Morteza Yazdani, Ali Ebadi Torkayesh, ?eljko Stević, Prasenjit Chatterjee, Sahand Asgharieh Ahari, and Violeta Doval Hernandez. An interval valued neutrosophic decision-making structure for sustainable supplier selection. *Expert Syst. Appl.*, 183:115354, 2021.

[287] Jyh-Cheng Yu and Kosuke Ishii. Design for robustness based on manufacturing variation patterns. *Journal of Mechanical Design*, 120:196–202, 1998.

[288] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[289] Lotfi A Zadeh. Biological application of the theory of fuzzy sets and systems. In *The Proceedings of an International Symposium on Biocybernetics of the Central Nervous System*, pages 199–206. Little, Brown and Comp. London, 1969.

[290] Lotfi A Zadeh. A fuzzy-set-theoretic interpretation of linguistic hedges. 1972.

[291] Lotfi A Zadeh. Fuzzy sets and their application to pattern classification and clustering analysis. In *Classification and clustering*, pages 251–299. Elsevier, 1977.

[292] Lotfi A Zadeh. Fuzzy logic, neural networks, and soft computing. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, pages 775–782. World Scientific, 1996.

[293] Lotfi A. Zadeh. Fuzzy topology. i. neighborhood structure of a fuzzy point and moore-smith convergence. 2003.

[294] Mohamed Bisher Zeina. Neutrosophic event-based queueing model. *International Journal of Neutrosophic Science*, 6(1):48–55, 2020.

[295] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and N. Chawla. Heterogeneous graph neural network. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[296] Xiaohong Zhang, Florentin Smarandache, and Xingliang Liang. Neutrosophic duplet semi-group and cancellable neutrosophic triplet groups. *Symmetry*, 9:275, 2017.

[297] Hua Zhao, Zeshui Xu, Shousheng Liu, and Zhong Wang. Intuitionistic fuzzy mst clustering algorithms. *Computers & Industrial Engineering*, 62(4):1130–1140, 2012.

[298] Liang Zhao, Shaocheng Qu, W. Zhang, and Zhili Xiong. An energy-saving fuzzy control system for highway tunnel lighting. *Optik*, 2019.

[299] Wufan Zhao, Claudio Persello, and Alfred Stein. Extracting planar roof structures from very high resolution images using graph neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 187:34–45, 2022.

[300] Ziwei Zheng, Huizhi Liang, Vaclav Snasel, Vito Latora, Panos Pardalos, Giuseppe Nicosia, and Varun Ojha. On learnable parameters of optimal and suboptimal deep learning models. *arXiv preprint arXiv:2408.11720*, 2024.

[301] Enwang Zhou and Alireza Khotanzad. Fuzzy classifier design using genetic algorithms. *Pattern Recognition*, 40(12):3401–3414, 2007.